

# The influence function of penalized regression estimators

Öllerer V, Croux C, Alfons A.



# The Influence Function of Penalized Regression Estimators

Viktoria Öllerer <sup>a\*\*</sup>, Christophe Croux <sup>a</sup> and Andreas Alfons <sup>b</sup>

<sup>a</sup>*Faculty of Economics and Business, KU Leuven, Belgium;*

<sup>b</sup>*Erasmus School of Economics, Erasmus University Rotterdam, The Netherlands*

To perform regression analysis in high dimensions, lasso or ridge estimation are a common choice. However, it has been shown that these methods are not robust to outliers. Therefore, alternatives as penalized M-estimation or the sparse least trimmed squares (LTS) estimator have been proposed. The robustness of these regression methods can be measured with the influence function. It quantifies the effect of infinitesimal perturbations in the data. Furthermore it can be used to compute the asymptotic variance and the mean squared error. In this paper we compute the influence function, the asymptotic variance and the mean squared error for penalized M-estimators and the sparse LTS estimator. The asymptotic biasedness of the estimators make the calculations non standard. We show that only M-estimators with a loss function with a bounded derivative are robust against regression outliers. In particular, the lasso has an unbounded influence function.

*Keywords:* Influence function; Lasso; Least Trimmed Squares; penalized M-regression; Sparseness

*AMS Subject Classification:* 62J20; 62J07

---

<sup>\*\*</sup>viktoria.oellerer@kuleuven.be

# 1 Introduction

Consider the usual regression situation. We have data  $(X, \mathbf{y})$ , where  $X \in \mathbb{R}^{n \times p}$  is the predictor matrix and  $\mathbf{y} \in \mathbb{R}^n$  the response vector. A linear model is commonly fit using least squares (LS) regression. It is well known that the LS estimator suffers from large variance in presence of high multicollinearity among the predictors. To overcome these problems, ridge [see Hoerl and Kennard, 1977] and lasso estimation [see Tibshirani, 1996] add a penalty term to the objective function of LS regression

$$\hat{\beta}_{LASSO} = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i' \beta)^2 + 2\lambda \sum_{j=1}^p |\beta_j| \quad (1)$$

$$\hat{\beta}_{RIDGE} = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n (y_i - \mathbf{x}_i' \beta)^2 + 2\lambda \sum_{j=1}^p \beta_j^2. \quad (2)$$

In contrast to the ridge estimator that only shrinks the coefficients of  $\hat{\beta}_{LS}$ , the lasso estimator also sets many of the coefficients to zero. This increases interpretability, especially in high-dimensional models. Both the ridge and the lasso estimator are special cases of a more general estimator, the penalized M-estimator [see Li et al., 2011]

$$\hat{\beta}_M = \arg \min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^n \rho(y_i - \mathbf{x}_i' \beta) + 2\lambda \sum_{j=1}^p J(\beta_j), \quad (3)$$

with loss function  $\rho : \mathbb{R} \rightarrow \mathbb{R}$  and penalty function  $J : \mathbb{R} \rightarrow \mathbb{R}$ .

The main drawback of the lasso is that it is not robust to outliers. As Alfons et al. [2013] have shown, the breakdown point of the lasso is  $1/n$ . This means that only one single outlier can make the estimate completely unreliable.

Hence, robust alternatives have been proposed. The least absolute deviation (LAD) estimator is well suited for heavy-tailed error distributions, but does not perform any variable selection. To simultaneously perform robust parameter estimation and variable selection Wang et al. [2007] combined LAD regression with lasso regression to LAD-lasso regression. However, this method has a finite sample breakdown point of  $1/n$  [see Alfons et al., 2013] and is thus not robust. Therefore Arslan [2012] provided a weighted version of the LAD-lasso that is made resistant to outliers by downweighting leverage points.

A popular robust estimator is the least trimmed squares (LTS) estimator [see Rousseeuw and Leroy, 1987]. Although its simple definition and fast computation make it interesting for practical application, it cannot be computed for high-dimensional data ( $p > n$ ).

Combining the lasso estimator with the LTS estimator, Alfons et al. [2013] developed the sparse LTS-estimator

$$\hat{\boldsymbol{\beta}}_{spLTS} = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \frac{1}{h} \sum_{i=1}^h r_{(i)}^2(\boldsymbol{\beta}) + \lambda \sum_{j=1}^p |\beta_j|, \quad (4)$$

where  $r_i^2(\boldsymbol{\beta}) = (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2$  denotes the squared residuals and  $r_{(1)}^2(\boldsymbol{\beta}) \leq \dots \leq r_{(n)}^2(\boldsymbol{\beta})$  their order statistics. Here  $\lambda \geq 0$  is a penalty parameter and  $h \leq n$  the size of the subsample that is considered to consist of non-outlying observations. This estimator can be applied to high-dimensional data with good prediction performance and high robustness. It also has a high breakdown point [see Alfons et al., 2013].

The rest of the paper is organized as follows. In Section 2, we define the penalized M-estimator at a functional level. In Section 3, we study its bias for different penalties and loss functions. We also give an explicit solution for sparse LTS for simple regression. In Section 4 we derive the influence function of the penalized M-estimator. Section 5 is devoted to the lasso. We give its influence function and describe the lasso as a limit case of penalized M-estimators with a differentiable penalty function. For sparse LTS we give the corresponding influence function in Section 6. In Section 7 we compare the plots of influence functions varying loss functions and penalties. A comparison at sample level is provided in Section 8. Using the results of Sections 4 - 6, Section 9 compares sparse LTS and different penalized M-estimators by looking at asymptotic variance and mean squared error. Section 10 concludes. The Appendix contains all proofs.

## 2 Functionals

Throughout the paper we work with the typical regression model

$$y = \mathbf{x}' \boldsymbol{\beta}_0 + e \quad (5)$$

with centered and symmetrically distributed error term  $e$ . We assume independence of the regressor  $\mathbf{x}$  and the error term  $e$ . We will denote the joint model distribution of  $\mathbf{x}$  and  $y$  by  $F$  and the covariance matrix of the regressors by  $\Sigma$ . The number of predictor variables is  $p$  and the variance of the error term  $e$  is denoted by  $\sigma^2$ .

The penalized M-estimator in (3) is defined at the sample level. The corresponding

definition at the population level, with  $(\mathbf{x}, y) \sim H$ , is

$$\boldsymbol{\beta}_M(H) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \mathbb{E}_H [\rho(y - \mathbf{x}'\boldsymbol{\beta})] + 2\lambda \sum_{j=1}^p J(\beta_j) \quad (6)$$

An example of a penalized M-estimator is the ridge functional, for which  $\rho(z) = J(z) = z^2$ . Also the lasso functional

$$\boldsymbol{\beta}_{LASSO}(H) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \left( \mathbb{E}_H [(y - \mathbf{x}'\boldsymbol{\beta})^2] + 2\lambda \sum_{i=1}^p |\beta_i| \right). \quad (7)$$

can be seen as a special case of the penalized M-estimator. However, its penalty is not differentiable, which will cause problems in the computation of the influence function.

To create more robust functionals, different loss functions than the classical quadratic loss function  $\rho(z) = z^2$  can be considered. Popular choices are the Huber function

$$\rho_H(z) = \begin{cases} z^2 & \text{if } |z| \leq k_H \\ 2k_H|z| - k_H^2 & \text{if } |z| > k_H \end{cases} \quad (8)$$

and Tukey's biweight function

$$\rho_{BI}(z) = \begin{cases} 1 - (1 - (\frac{z}{k_{BI}})^2)^3 & \text{if } |z| \leq k_{BI} \\ 1 & \text{if } |z| > k_{BI}. \end{cases} \quad (9)$$

The Huber loss function  $\rho_H$  is a continuous, differentiable function that is quadratic in a central region  $[-k_H, k_H]$  and increases only linearly outside of this interval (compare Figure 1). Therefore it gives less weight to extreme residuals than the quadratic loss, while it is still differentiable at zero in contrast to an absolute value loss. The main advantage of the biweight function  $\rho_{BI}$  (sometimes also called 'bisquared' function) is that it is a smooth function that cuts off large values. Small residuals are weighted similarly as with a quadratic loss function, whereas large residuals are trimmed (compare Figure 1). The choice of the tuning constants  $k_{BI}$  and  $k_H$  determines the breakdown point and efficiency of the functionals. We use  $k_{BI} = 4.685$  and  $k_H = 1.345$ , which gives 95% of efficiency for a standard normal error distribution in the unpenalized case.

Another possible choice of a penalty function is the smoothly clipped absolute deviation (SCAD) penalty [see Fan and Li, 2001]

$$J_{SCAD}(\beta) = \begin{cases} |\beta| & \text{if } |\beta| \leq \lambda \\ -\frac{(|\beta| - a\lambda)^2}{2(a-1)\lambda} + \lambda \frac{a+1}{2} & \text{if } \lambda < |\beta| \leq a\lambda \\ \lambda \frac{a+1}{2} & \text{if } |\beta| > a\lambda. \end{cases} \quad (10)$$

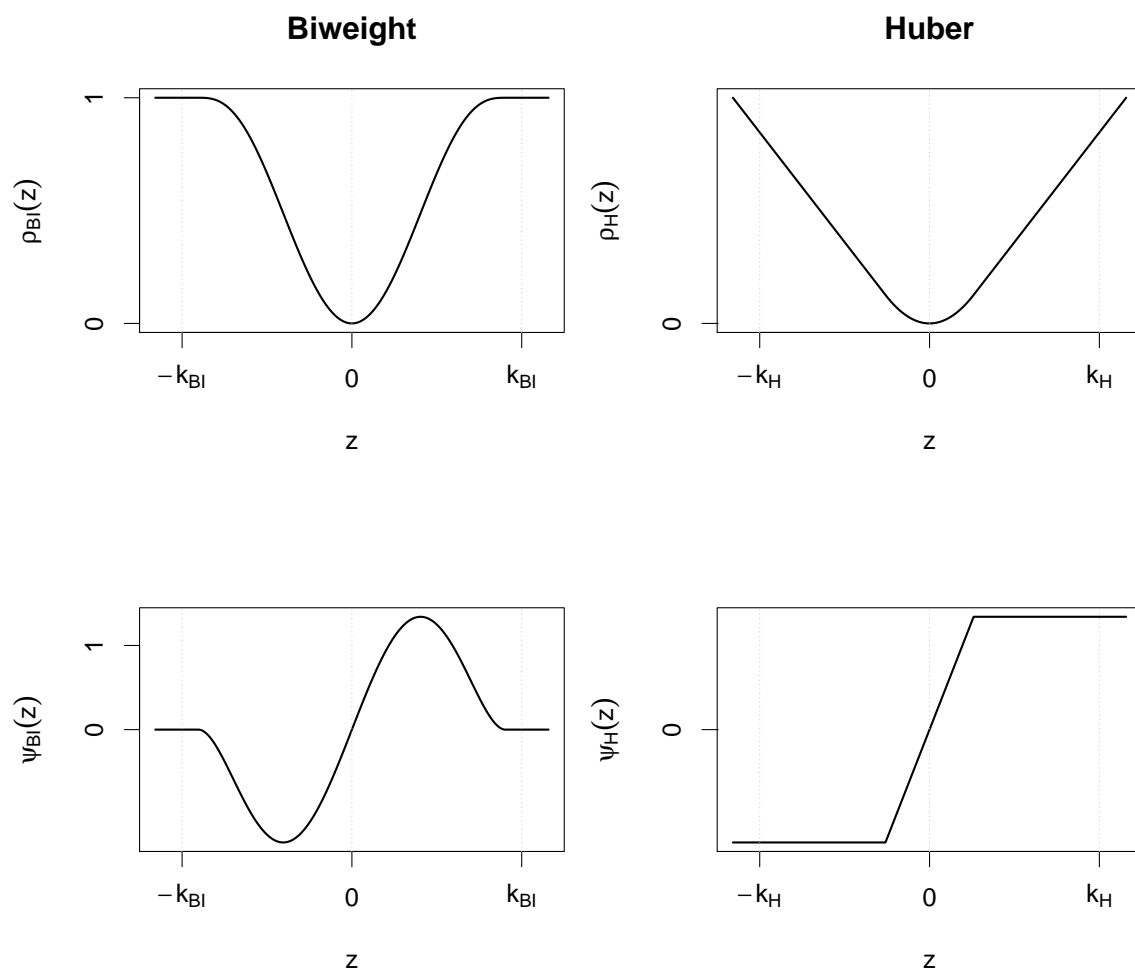


Figure 1: Biweight and Huber loss function and their first derivatives. The first derivative determines the weight an observations gets when computing the estimator.

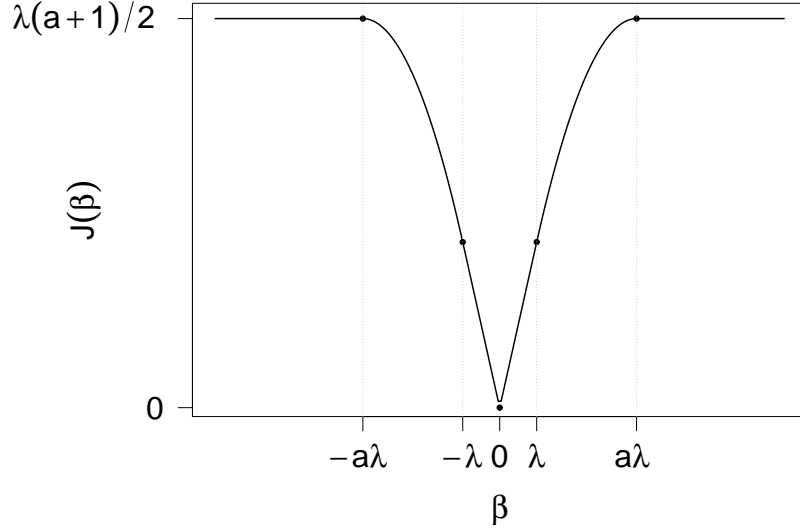


Figure 2: The smoothly clipped absolute deviation (SCAD) penalty function

While the SCAD functional, exactly as the lasso, shrinks (with respect to  $\lambda$ ) small parameters to zero, large values are not shrunk at all, exactly as in LS regression.

The definition of the sparse LTS estimator at a population level is

$$\boldsymbol{\beta}_{spLTS}(H) = \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^p} \mathbb{E}_H[(y - \mathbf{x}'\boldsymbol{\beta})^2 I_{|y - \mathbf{x}'\boldsymbol{\beta}| \leq q_\beta}] + \alpha \lambda \sum_{j=1}^p |\beta_j|, \quad (11)$$

with  $q_\beta$  the  $\alpha$ -quantile of  $|y - \mathbf{x}'\boldsymbol{\beta}| \sim \bar{H}^\beta$ , when  $(\mathbf{x}, y) \sim H$ . As recommended in Alfons et al. [2013], we take  $\alpha = 0.75$ . The major difference to the penalized M-functionals is that the loss function of sparse LTS  $\rho(z) = z^2 I_{|z| \leq q_z}$  is not continuous. Besides, the cutoff-point  $q_z$  is not fixed and depends on the distribution of  $z$ .

### 3 Bias

In contrast to the LS functional, the penalized M-functional  $\boldsymbol{\beta}_M$  has a bias at the model distribution

$$\text{Bias}(\boldsymbol{\beta}_M, F) = \boldsymbol{\beta}_M(F) - \boldsymbol{\beta}_0.$$

The difficulty here lies in the computation of the functional  $\beta_M(F)$ . For the lasso functional, there exists an explicit solution only for simple regression (i.e.  $p = 1$ )

$$\beta_{LASSO}(H) = \text{sign}(\beta_{LS}(H)) \left( |\beta_{LS}(H)| - \frac{\lambda}{\mathbb{E}_H[x^2]} \right)_+ . \quad (12)$$

Here  $\beta_{LS}(H) = \mathbb{E}_H[xy]/\mathbb{E}_H[x^2]$  denotes the least squares functional and  $(z)_+ = \max(0, z)$  the positive part function. For completeness, we give a proof of Equation (12) in the Appendix. For multiple regression the lasso functional at the model distribution  $F$  can be computed using the idea of the coordinate descent algorithm (see Section 5), with the model parameter  $\beta_0$  as a starting value. Similarly, also for the SCAD functional there exists an explicit solution only for simple regression

$$\beta_{SCAD}(H) = \begin{cases} (|\beta_{LS}(H)| - \frac{\lambda}{\mathbb{E}_F[x^2]})_+ \text{sign}(\beta_{LS}(H)) & \text{if } |\beta_{LS}(H)| \leq \lambda + \frac{\lambda}{\mathbb{E}_F[x^2]} \\ \frac{(a-1)\mathbb{E}_F[x^2]\beta_{LS}(H) - a\lambda \text{sign}(\beta_{LS}(H))}{(a-1)\mathbb{E}_F[x^2] - 1} & \text{if } \lambda + \frac{\lambda}{\mathbb{E}_F[x^2]} < |\beta_{LS}(H)| \leq a\lambda \\ \beta_{LS}(H) & \text{if } |\beta_{LS}(H)| > a\lambda. \end{cases} \quad (13)$$

This can be proved using the same ideas as in the computation of the solution for the lasso functional in simple regression (see Proof of Equation (12) in the appendix). Here the additional assumption  $\mathbb{E}_H[x^2] > 1/(a-1)$  is needed. As can be seen from Equation (13), the SCAD functional is unbiased at the model  $F$  for large values of the parameter  $\beta_0$ .

To compute the value of a penalized M-functional that does not use a quadratic loss function, the iteratively reweighted least squares (IRLS) algorithm [see Osborne, 1985] can be adapted to find a solution, with weights depending on the loss function used. The estimator used in every reweighting step then depends on the loss function used: instead of weighted lasso, weighted SCAD (both computed via coordinate descent) or weighted ridge (computed via explicit solution) are possibilities. In the weights, expected values depending on the current estimate and the distribution  $H$  will show up. These expected values are best simulated through Monte Carlo. (We used  $10^5$  replications.)

For sparse LTS we can again find an explicit solution for simple regression with normal predictor and error term.

**Lemma 1.** *Let  $y = x\beta + e$  be an simple regression model as in (5). If  $x$  and  $e$  are normally distributed, the explicit solution of the sparse LTS functional (11) is*

$$\beta_{spLTS}(F) = \text{sign}(\beta_0) \left( |\beta_0| - \frac{\alpha\lambda}{2c_1\mathbb{E}_F[x^2]} \right)_+ \quad (14)$$



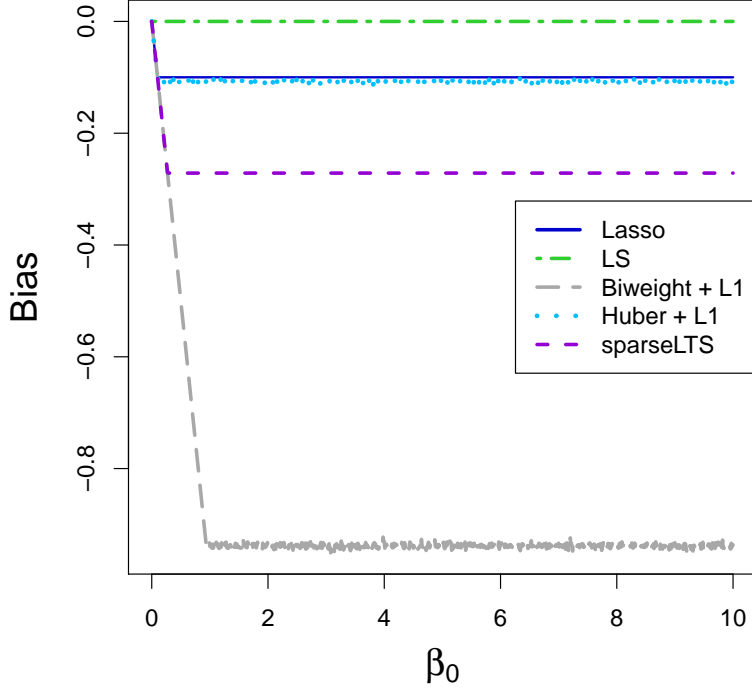


Figure 3: Bias of various functionals for different values of  $\beta_0$  ( $\lambda = 0.1$  fixed). Note that the small fluctuations are due to Monte Carlo simulations in the computation of the functional.

with  $c_1 = \alpha - 2q_\alpha\phi(q_\alpha)$ ,  $q_\alpha$  the  $\frac{\alpha+1}{2}$ -quantile of the standard normal distribution and  $\phi$  its density.

To study the bias of the various functionals of Section 2, we take  $p = 1$  and assume  $x$  and  $e$  as standard normally distributed. We use  $\lambda = 0.1$  for all functionals. Figure 3 displays the bias as a function of  $\beta_0$ . Please note that this is not a comparison of different loss function, but should only demonstrate the bias of each functional individually. Of all functionals used only LS has a zero bias.  $L_1$  penalized functionals have a constant bias for values of  $\beta_0$  that are not shrunk to zero. For smaller values of  $\beta_0$  the bias increases monotonously in absolute value.

## 4 The Influence Function

The robustness of a functional  $\beta$  can be measured via the influence function

$$IF((\mathbf{x}_0, y_0), \beta, H) = \left. \frac{\partial}{\partial \epsilon} \left[ \beta((1 - \epsilon)H + \epsilon \delta_{(\mathbf{x}_0, y_0)}) \right] \right|_{\epsilon=0}.$$

It describes the effect of infinitesimal, pointwise contamination in  $(\mathbf{x}_0, y_0)$  on the functional  $\beta$ . Here  $H$  denotes any distribution and  $\delta_{\mathbf{z}}$  the point mass distribution at  $\mathbf{z}$ . To compute the influence function of the penalized M-functional (6), smoothness conditions for functions  $\rho(\cdot)$  and  $J(\cdot)$  have to be assumed.

**Proposition 1.** *Let  $y = \mathbf{x}'\beta_0 + e$  be a regression model as defined in (5). Furthermore, let  $\rho, J : \mathbb{R} \rightarrow \mathbb{R}$  be twice differentiable functions and denote the derivative of  $\rho$  by  $\psi := \rho'$ . Then the influence function  $IF$  of the penalized M-functional  $\beta_M$  for  $\lambda \geq 0$  is given by*

$$\begin{aligned} IF((\mathbf{x}_0, y_0), \beta_M, F) = & \\ & = (\mathbb{E}_F[\psi'(y - \mathbf{x}'\beta_M(F))\mathbf{x}\mathbf{x}'] + 2\lambda \text{diag}(J''(\beta_M(F))))^{-1} \cdot \\ & \cdot (\psi(y_0 - \mathbf{x}_0'\beta_M(F))\mathbf{x}_0 - \mathbb{E}_F[\psi(y - \mathbf{x}'\beta_M(F))\mathbf{x}]). \end{aligned} \quad (15)$$

The influence function (15) of the penalized M-functional is unbounded in  $\mathbf{x}_0$  and is only bounded in  $y_0$  if  $\psi(\cdot)$  is bounded. In Section 7 we will see that the effect of the penalty on the shape of the influence function is quite small compared to the effect of the loss function.

As the ridge functional can be seen as a special case of the penalized M-functional (6), its influence function follows as a corollary:

**Corollary 1.** *The influence function of the ridge functional  $\beta_{\text{RIDGE}}$  is*

$$IF((\mathbf{x}_0, y_0), \beta_{\text{RIDGE}}, F) = (\mathbb{E}_F[\mathbf{x}\mathbf{x}'] + 2\lambda I_p)^{-1} \left( (y_0 - \mathbf{x}_0'\beta_{\text{RIDGE}}(F))\mathbf{x}_0 + \mathbb{E}_F[\mathbf{x}\mathbf{x}'] \text{Bias}(\beta_{\text{RIDGE}}, F) \right). \quad (16)$$

As the function  $\psi(z) = 2z$  is unbounded, the influence function (16) of the ridge functional is unbounded. Thus, the ridge functional is not robust to any kind of outliers.

The penalty function  $J(z) := |z|$  of the lasso functional and the sparse LTS functional is not twice differentiable at zero. Therefore, those functionals are no special cases of the M-functional used in Proposition 1 and have to be considered separately to derive the influence function.

## 5 The Influence Function of the Lasso

For simple regression, i.e. for  $p = 1$ , an explicit solution for the lasso functional exists, see Equation (12). With that the influence function can be computed easily.

**Lemma 2.** *Let  $y = x\beta_0 + e$  be an simple regression model as in (5). Then the influence function IF of the lasso functional is*

$$IF((x_0, y_0), \beta_{LASSO}, F) = \begin{cases} 0 & \text{if } -\frac{\lambda}{\mathbb{E}_F[x^2]} \leq \beta_0 < \frac{\lambda}{\mathbb{E}_F[x^2]} \\ \frac{x_0(y_0 - \beta_0 x_0)}{\mathbb{E}_F[x^2]} - \lambda \frac{\mathbb{E}_F[x^2] - x_0^2}{(\mathbb{E}_F[x^2])^2} \text{sign}(\beta_0) & \text{otherwise.} \end{cases} \quad (17)$$

Similar to the influence function of the ridge functional (16), the IF of the lasso functional (17) is unbounded in both variables  $x_0$  and  $y_0$  in case the coefficient  $\beta_{LASSO}$  is not shrunk to zero (case 2 in equation (17)). Otherwise the IF is constantly zero. The reason of the similarity of the IF of the lasso and the ridge functional is that both are a shrunken version of the LS functional.

As there is no explicit solution in multiple regression for the lasso functional, its influence function cannot be computed easily. However, Friedman et al. [2007] and Fu [1998] found an algorithm, the *coordinate descent algorithm* (also *shooting algorithm*), to split up the multiple regression into a number of simple regressions. The idea of the coordinate descent algorithm at population level is to compute the lasso functional (7) variable by variable. Repeatedly, one variable  $j \in \{1, \dots, p\}$  is selected. The value of the functional  $\beta_j^{cd}$  is then computed holding all other coefficients  $k \neq j$  fixed at their previous value  $\beta_k^*$

$$\begin{aligned} \beta_j^{cd}(H) &= \arg \min_{\beta_j \in \mathbb{R}} \mathbb{E}_H[(y - \sum_{k \neq j} x_k \beta_k^* - x_j \beta_j)^2] + 2\lambda \sum_{k \neq j} |\beta_k^*| + 2\lambda |\beta_j| \\ &= \arg \min_{\beta_j \in \mathbb{R}} \mathbb{E}_H[(y - \sum_{k \neq j} x_k \beta_k^* - x_j \beta_j)^2] + 2\lambda |\beta_j|. \end{aligned} \quad (18)$$

This can be seen as simple lasso regression with partial residuals  $y - \sum_{k \neq j} x_k \beta_k^*$  as response and the  $j$ th coordinate  $x_j$  as covariate. Thus, the new value of  $\beta_j^{cd}(H)$  can be easily computed using Equation (12). Looping through all variables repeatedly, convergence to the lasso functional (7) will be reached for any starting value [see Friedman et al., 2007; Tseng, 2001].

For the coordinate descent algorithm an influence function can be computed similarly as for simple regression. However, now the influence function depends on the influence function of the previous value  $\beta^*$ .

**Lemma 3.** *Let  $y = \mathbf{x}'\beta_0 + e$  be the regression model of (5). Then the influence function IF of the  $j$ th coordinate of the lasso functional (18) computed via coordinate descent is*

$$IF((\mathbf{x}_0, y_0), \beta_j^{cd}, F) = \begin{cases} 0 & \text{if } |\mathbb{E}_F[x_j \tilde{y}^{(j)}]| < \lambda \\ \frac{-\mathbb{E}_F[x_j \mathbf{x}^{(j)'}] IF((\mathbf{x}_0, y_0), \beta^{*(j)}, F) + (y_0 - \mathbf{x}_0^{(j)'} \beta^{*(j)}(F))(\mathbf{x}_0)_j - \frac{\mathbb{E}_F[x_j \tilde{y}^{(j)}](\mathbf{x}_0)_j^2}{(\mathbb{E}_F[x_j^2])^2}}{\mathbb{E}_F[x_j^2]} & \text{otherwise} \\ -\lambda \frac{\mathbb{E}_F[x_j^2] - (\mathbf{x}_0)_j^2}{(\mathbb{E}_F[x_j^2])^2} \text{sign}(\mathbb{E}_F[x_j \tilde{y}^{(j)}]) & \end{cases} \quad (19)$$

where for any vector  $\mathbf{z}$  we define  $\mathbf{z}^{(j)} = (z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_p)'$ ,  $\tilde{y}^{(j)} := y - \mathbf{x}^{(j)'} \beta^{*(j)}(F)$ , with  $\beta^{*(j)}$  the functional representing the value of the coordinate descent algorithm at population level in the previous step.

To obtain a formula for the influence function of the lasso functional in multiple regression, we can use the result of Lemma 3. The following proposition holds.

**Proposition 2.** *Let  $y = \mathbf{x}'\beta_0 + e$  be the regression model of (5). Without loss of generality let  $\beta_{LASSO}(F) = ((\beta_{LASSO}(F))_1, \dots, (\beta_{LASSO}(F))_k, 0, \dots, 0)'$  with  $k \leq p$  and  $(\beta_{LASSO}(F))_j \neq 0 \forall j = 1, \dots, k$ . Then the influence function IF of the lasso functional (7) is*

$$IF((\mathbf{x}_0, y_0), \beta_{LASSO}, F) = \begin{pmatrix} (\mathbb{E}_F[\mathbf{x}_{1:k} \mathbf{x}'_{1:k}])^{-1} \left( (\mathbf{x}_0)_{1:k} (y_0 - \mathbf{x}_0' \beta_{LASSO}(F)) - \mathbb{E}_F[\mathbf{x}_{1:k} (y - \mathbf{x}' \beta_{LASSO}(F))] \right) \\ \mathbf{0}_{p-k} \end{pmatrix} \quad (20)$$

with the notation  $\mathbf{z}_{r:s} = (z_r, z_{r+1}, \dots, z_{s-1}, z_s)'$  for  $\mathbf{z} \in \mathbb{R}^p$ ,  $r, s \in \{1, \dots, p\}$  and  $r \leq s$ .

Another approach to compute the influence function of the lasso functional is to consider it as a limit case of functionals satisfying the conditions of Proposition 1. The following sequence of hyperbolic tangent functions converges to the sign-function

$$\lim_{K \rightarrow \infty} \tanh(Kx) = \begin{cases} +1 & \text{if } x > 0 \\ -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0. \end{cases}$$

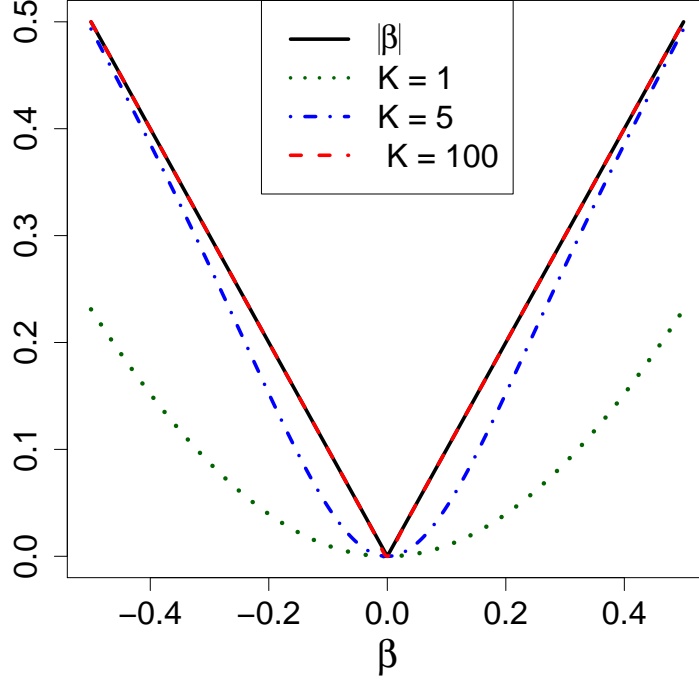


Figure 4: Approximation of  $|\beta|$  using  $\beta \cdot \tanh(K\beta)$

Hence, it can be used to get a smooth approximation of the absolute value function

$$|x| = x \cdot \text{sign}(x) = \lim_{K \rightarrow \infty} x \cdot \tanh(Kx). \quad (21)$$

The larger the value of  $K > 1$ , the better the approximation becomes (see Figure 4). Therefore the penalty function  $J_K(\beta_j) = \beta_j \tanh(K\beta_j)$  is an approximation of  $J_{LASSO}(\beta_j) = |\beta_j|$ . As  $J_K$  is a smooth function, the influence function of the corresponding functional

$$\beta_K(F) = \arg \min_{\beta \in \mathbb{R}^p} \mathbb{E}_F[(y - \mathbf{x}'\beta)^2] + 2\lambda \sum_{j=1}^p J_K(\beta_j). \quad (22)$$

can be computed by applying Proposition 1. Taking the limit of this influence function, we get the influence function of the lasso functional. It coincides with the expression given in Proposition 2.

**Lemma 4.** *Let  $y = \mathbf{x}'\beta_0 + e$  be the regression model of (5). Without loss of generality let  $\beta_{LASSO}(F) = ((\beta_{LASSO}(F))_1, \dots, (\beta_{LASSO}(F))_k, 0, \dots, 0)'$  with  $k \leq p$  and  $(\beta_{LASSO}(F))_j \neq 0 \forall j = 1, \dots, k$ . Then the influence function  $IF$  of the penalized  $M$ -estimator (22) converges to the  $IF$  of the lasso functional given in (20) as  $K$  converges to infinity.*

## 6 The Influence Function of sparse LTS

For sparse LTS, computation of the influence function is more difficult than for the lasso. In addition to the nondifferentiable penalty function, sparse LTS also has a discontinuous loss function. For simplicity, we therefore assume a univariate normal distribution for the predictor  $x$  and the error  $e$ .

**Lemma 5.** *Let  $y = x\beta_0 + e$  be a simple regression model as in (5). If  $x$  and  $e$  are normally distributed, the influence function  $IF$  of the sparse LTS functional (14) is*

$$IF((x_0, y_0), \beta_{spLTS}, F) = \begin{cases} 0 & \text{if } -\frac{\alpha\lambda}{2c_1\mathbb{E}_F[x^2]} < \beta_0 \leq \frac{\alpha\lambda}{2c_1\mathbb{E}_F[x^2]} \\ (\beta_{spLTS}(F) - \beta_0) - \frac{q_\alpha^2(I_{[|r_0| \leq q_\alpha]} - \alpha)(\beta_0 - \beta_{spLTS}(F))}{\alpha - 2q_\alpha\phi(q_\alpha)} + \\ + \frac{x_0(y_0 - x_0\beta_{spLTS}(F))I_{[|r_0| \leq q_\alpha]}}{(\alpha - 2q_\alpha\phi(q_\alpha))\mathbb{E}_F[x^2]} & \text{otherwise.} \end{cases} \quad (23)$$

with  $r_0 = \frac{y_0 - x_0\beta_{spLTS}(F)}{\sqrt{\sigma^2 + (\beta_0 - \beta_{spLTS}(F))^2\mathbb{E}_F[x^2]}}$  and the same notation as in Lemma 1.

Lemma 5 shows that the influence function of the sparse LTS functional may become unbounded for points  $(x_0, y_0)$  that follow the model, i.e. for good leverage points, but remains bounded elsewhere, in particular for bad leverage points and vertical outliers. This shows the good robust properties of sparse LTS.

## 7 Plots of Influence Functions

We first compare the effects of different penalties and take a quadratic loss function. We consider least squares (LS), ridge and lasso regression as well as the SCAD penalty (10). To compute ridge and lasso regression a value for the penalty parameter  $\lambda$  is needed, and for SCAD another additional parameter  $a$  has to be specified. We choose a fixed value  $\lambda = 0.1$  and, as proposed by Fan and Li [2001], we use  $a = 3.7$ .

Influence functions can only be plotted for simple regression  $y = x\beta_0 + e$ , i.e. for  $p = 1$ . We specify the predictor and the error as independent and standard normally distributed. For the parameter  $\beta_0$  we use a parameter  $\beta_0 = 1.5$  that will not be shrunk to zero by any of the functionals, as well as  $\beta_0 = 0$  to focus also on the sparseness of the functionals. Figures 5 and 6 show the plots of the IFs for LS, ridge, lasso and SCAD for both values of  $\beta_0$ . Examining Figure 5, one could believe that all influence functions

are equal. The same applies for the IFs of LS and ridge in Figure 6. However, this is not the case. All IFs are different of one another because their bias and the second derivative of the penalty appear in the expression of the IF. Those terms are different for the different functionals. Usually, the differences are minor. Note, however, that for some specific choices of  $\lambda$  and  $\beta_0$  differences can be substantial. For  $\beta_0 = 0$ , see Figure 6, SCAD and lasso produce a constantly zero influence function. We may conclude that in most cases the effect of the penalty function on the shape of the IF is minor.

To compare different loss functions, we use Huber loss (8), biweight loss (9) and sparse LTS (11), each time combined with the  $L_1$ -penalty  $J(\beta) = |\beta|$  to achieve sparseness. For the simple regression model  $y = x\beta_0 + e$ , we specify the predictor and the error as independent and standard normally distributed and consider  $\beta_0 = 0$  and  $\beta_0 = 1.5$ . Furthermore, we fix  $\lambda = 0.04$ .

Figure 7 shows the IFs of these functionals with Huber and biweight loss function. They clearly differ from the ones using the classic quadratic loss for coefficients  $\beta_0$  that are not shrunk to zero (compare to panels corresponding to the lasso in Figures 6 and 5). The major difference is that the IFs of functionals with a bounded loss function (sparse LTS, biweight) are only unbounded for good leverage points and bounded for regression outliers. This indicates the robust behavior of the functionals. It is even further emphasized by the fact that those observations  $(x_0, y_0)$  with big influence are the ones with small residuals  $y_0 - x_0\beta_0$ , that is the ones that closely follow the underlying model distribution. Observations with large residuals have small and constant influence. In contrast, the unbounded Huber loss function does not achieve robustness against all types of outliers. Only for outliers in the response the influence is constant (for a fixed value of  $x_0$ ). However, if the predictor values increases, the influence of the corresponding observation increases linearly. For a quadratic loss function the increase would be quadratic. Thus, a Huber loss reduces the influence of bad leverage points, but does not bound it. For  $\beta(F) = 0$  and for all loss functions, the  $L_1$  penalized functionals produce a constantly zero IF, thus, creating sparseness also under small perturbation from the model. To sum up, a Huber loss function performs better than a quadratic loss, but both cannot bound the influence of bad leverage points. Only sparse LTS and the penalized M-functional with biweight loss are very robust. They are able to bound the impact of observations that lie far away from the model, while observations that closely follow the model get a very high influence.

We simulate the expected values that appear in the influence function (15) by Monte

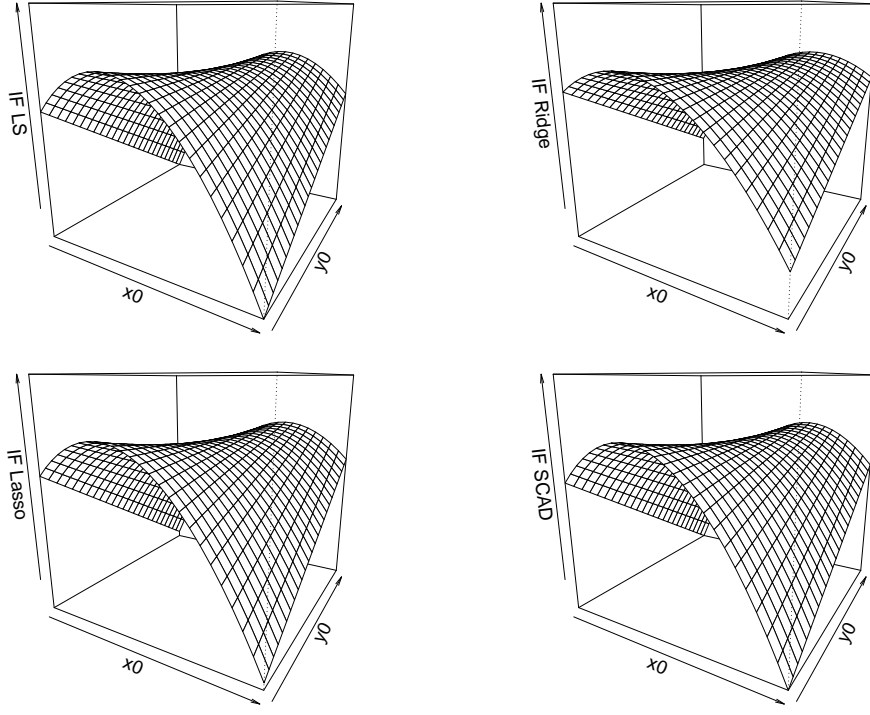


Figure 5: Influence functions for different penalty functions (least squares, ridge, lasso and SCAD) for  $\beta_0 = 1.5$  with  $(x_0, y_0) \in [-10, 10]^2$  and the vertical axis ranging from  $-250$  to  $100$

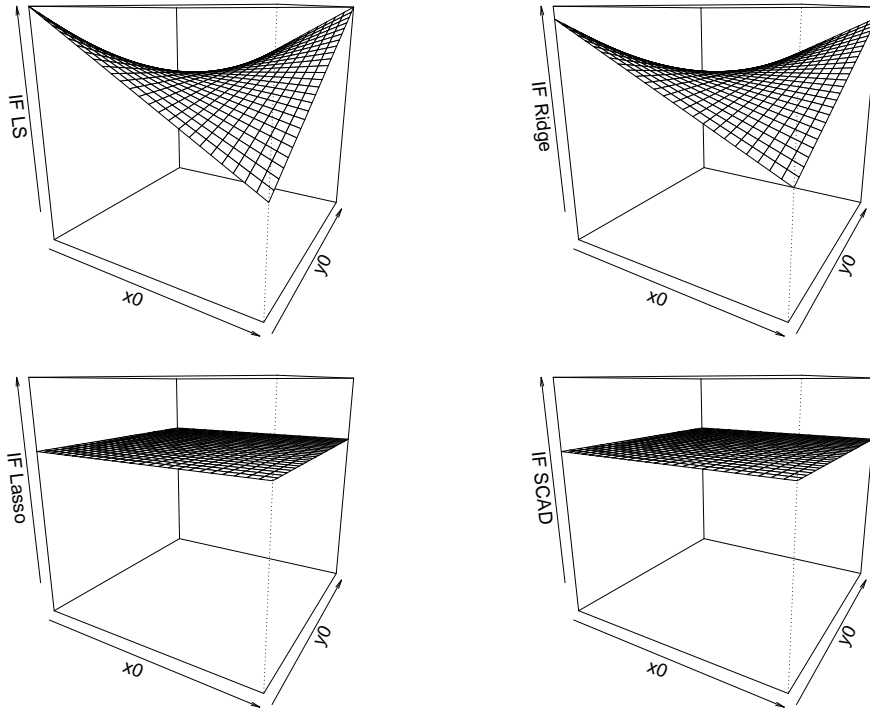


Figure 6: Influence functions for different penalty functions (least squares, ridge, lasso and SCAD) for  $\beta_0 = 0$  with  $(x_0, y_0) \in [-10, 10]^2$  and the vertical axis ranging from  $-250$  to  $100$



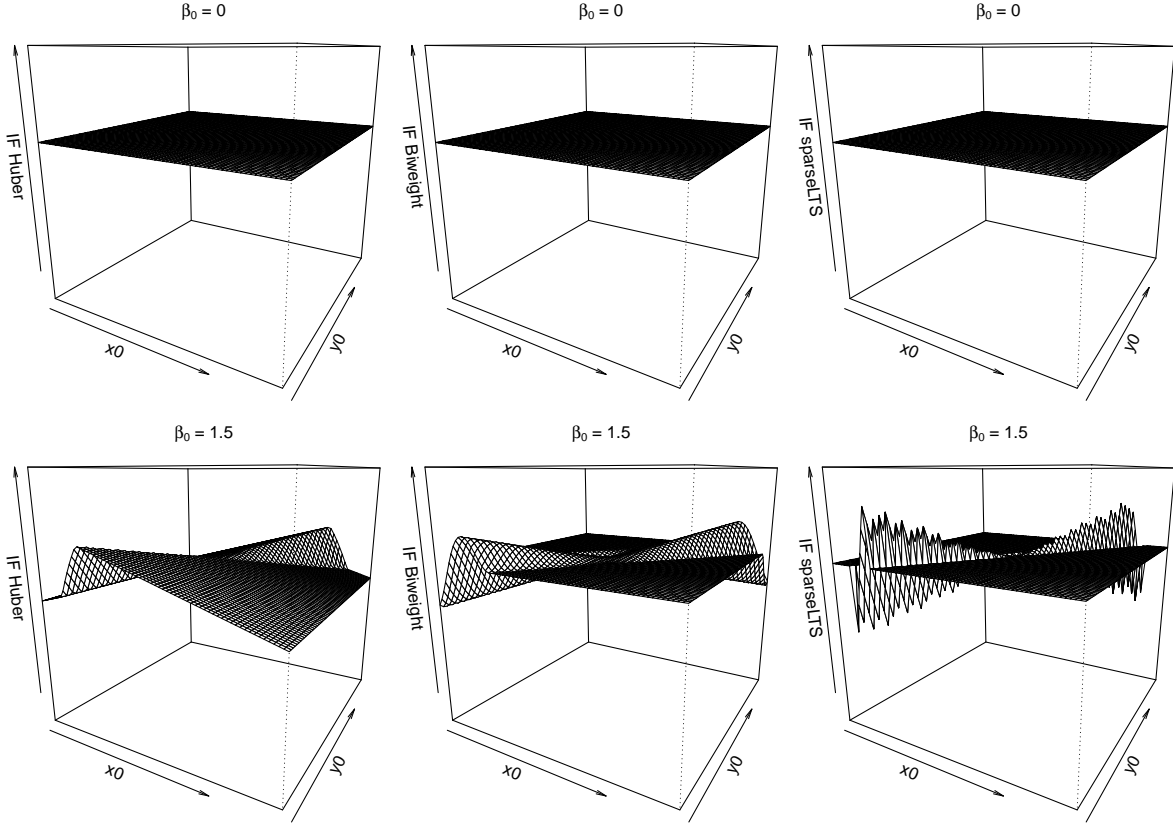


Figure 7: Influence functions for different loss functions (Huber, biweight, sparse LTS) and  $L_1$  penalty for  $\beta_0 = 0$  and  $\beta_0 = 1.5$  with  $(x_0, y_0) \in [-10, 10]^2$  and the vertical axis ranging from  $-75$  to  $40$

Carlo simulation (using  $10^5$  replications). Furthermore, Proposition 1 can actually not be applied as the lasso penalty is not differentiable. However, using either the *tanh* approximation (21) or the same approach as in the proof of Lemma 2, one can show that the IF of these functionals equals zero in case the functional equals zero and (15) otherwise.

## 8 Sensitivity Curves

To study the robustness of the different penalized M-estimators from Section 7 at sample level, we compute sensitivity curves [see Maronna et al., 2006], an empirical version of

the influence function. For an estimator  $\hat{\beta}$  and at sample  $(X, \mathbf{y})$ , it is defined as

$$SC(\mathbf{x}_0, y_0, \hat{\beta}) = \frac{\hat{\beta}(X \cup \{\mathbf{x}_0\}, \mathbf{y} \cup \{y_0\}) - \hat{\beta}(X, \mathbf{y})}{\frac{1}{n+1}}.$$

To compute the penalized estimators, we use the coordinate descent algorithm. As a starting value, we use the LS estimate for estimators using a quadratic loss, and the robust sparse LTS-estimate for the others. Sparse LTS can be easily and fast computed using the `sparseLTS` function of the R package `robustHD`. Furthermore, we divide the argument of the  $\rho$ -function in (3) by a preliminary scale estimate. For simplicity we use the MAD of the residuals of the initial estimator used in the coordinate descent algorithm.

Figures 8 and 9 show the sensitivity curves for estimators  $\hat{\beta}$  with quadratic loss function and the different penalties least squares, ridge, lasso and SCAD for parameters  $\beta_0 = 1.5$  and  $\beta_0 = 0$ , respectively. We can compare these figures to the theoretical influence functions in Figures 5 and 6. Examining Figure 8, we see that for  $\beta_0 = 1.5$ , the results match the theoretical ones. For  $\beta_0 = 0$ , see Figure 9, the sensitivity curve is again comparable to the influence function. For the lasso and SCAD a small deviation from the constantly zero sensitivity curve can be spotted in the left and right corner. This indicates that the number of observations  $n$  is too small to get the same results as at population level for observations  $(x_0, y_0)$  that lie far away from the model.

We also want to compare the results for estimators using different loss function. Therefore we look at sparse LTS and the  $L_1$ -penalized Huber- and biweight-M-estimators, as in Section 7. Their sensitivity curves are plotted in Figure 10. They resemble the shape of the influence functions in Figure 7.

To conclude, we may say that the sensitivity curves match the corresponding influence functions.

## 9 Asymptotic Variance and Mean Squared Error

We can also evaluate the performance of any functional  $T$  by the Asymptotic Variance (ASV), given by

$$ASV(T, H) = n \cdot \lim_{n \rightarrow \infty} \text{Var } T_n,$$

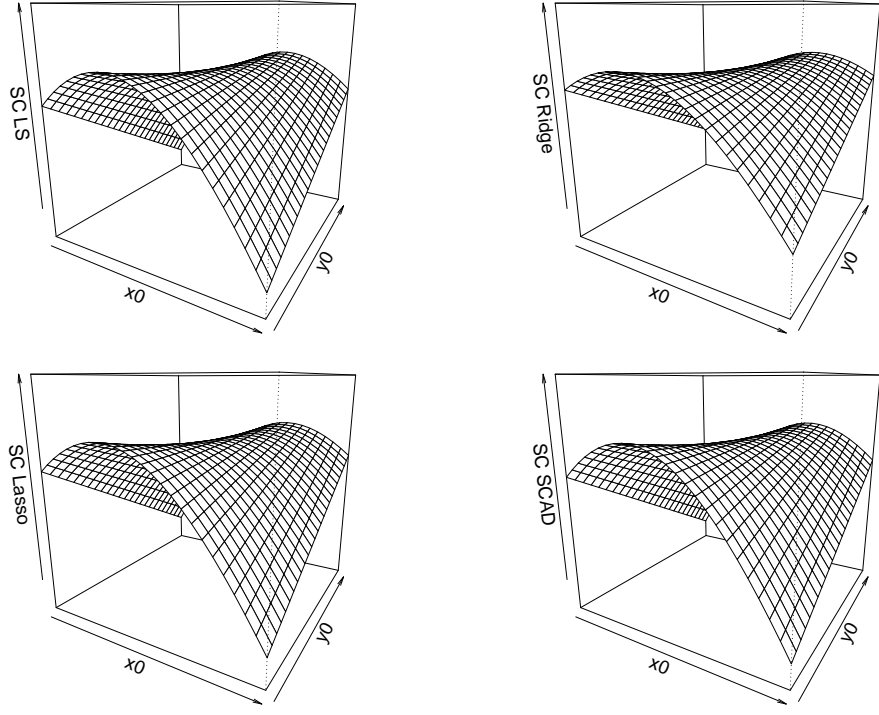


Figure 8: Sensitivity curve for different penalty functions (least squares, ridge, lasso and SCAD) for  $\beta_0 = 1.5$  with  $(x_0, y_0) \in [-10, 10]^2$  and the vertical axis ranging from  $-250$  to  $100$

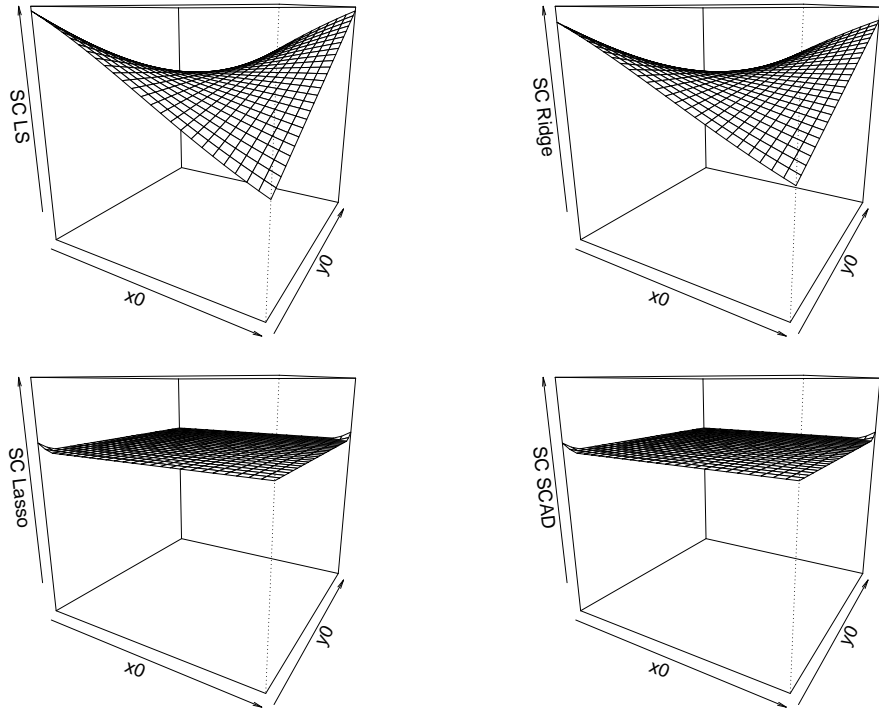


Figure 9: Sensitivity curve for different penalty functions (least squares, ridge, lasso and SCAD) for  $\beta_0 = 0$  with  $(x_0, y_0) \in [-10, 10]^2$  and the vertical axis ranging from  $-250$  to  $100$

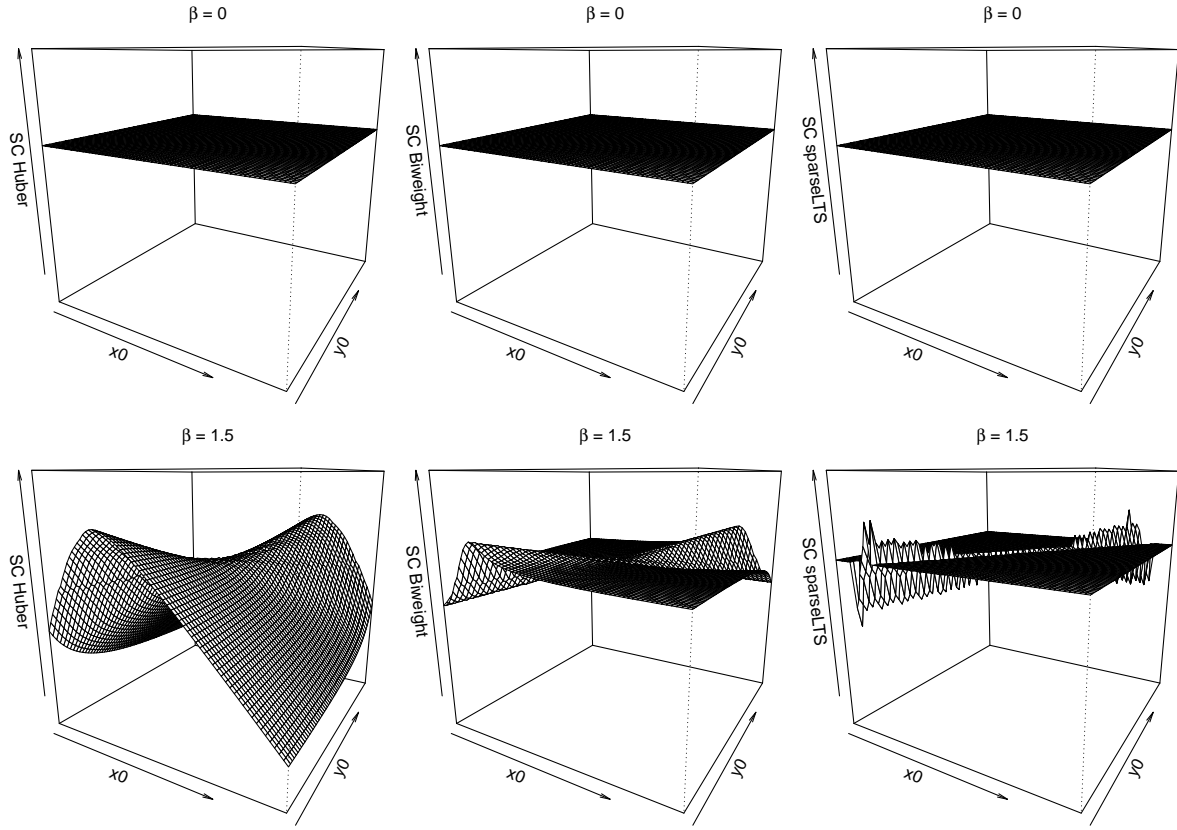


Figure 10: Sensitivity curve for different loss functions (Huber, biweight, sparse LTS) and  $L_1$  penalty for  $\beta_0 = 0$  and  $\beta_0 = 1.5$  with  $(x_0, y_0) \in [-10, 10]^2$  and the vertical axis ranging from  $-75$  to  $40$

where the estimator  $T_n$  is the functional  $T$  evaluated at the empirical distribution. The ASV can be explicitly computed using the influence function [see Hampel et al., 1986]

$$ASV(T, H) = \int IF((\mathbf{x}_0, y_0), T, H) \cdot IF((\mathbf{x}_0, y_0), T, H)' dH((\mathbf{x}_0, y_0)). \quad (24)$$

Using formulas of Sections 4 - 6, computation of the integral (24) is possible using Monte Carlo integration (we chose  $N = 10^4$  replications). We present results for simple regression as for multiple regression, the component-wise results match the results for simple regression.

Figure 11 shows the ASV of six different functionals (LS, lasso, ridge, biweight loss with  $L_1$  penalty, Huber loss with  $L_1$  penalty, sparse LTS) as a function of  $\lambda$  for  $\beta_0 = 1.5$ . As the ASV of LS is constantly one for any value  $\lambda$  and  $\beta_0$ , it is used as a reference point in all four panels. All sparse functionals show a jump to zero in their ASV after having increased quickly to their maximum. This is due to parameters estimated exactly zero, for values of  $\lambda$  sufficiently large. In the left upper panel the ASV of ridge is added. It is smaller than the ASV of LS and decreases monotonously to zero. Generally, for the optimal  $\lambda$ , LS has high ASV, ridge a reduced one. The smallest ASV can be achieved by the sparse functionals. But they can also get considerably high values for bad choices of  $\lambda$ . We omit the plots for  $\beta_0 = 0$  because the ASV of ridge behaves similarly as in Figure 11 and the ASV of the other, sparse functionals is constantly zero.

In general, robust functionals have a bias (see Section 3). Hence, considering only ASV is not sufficient to evaluate the precision of functionals. A more informative measure is the Mean Squared Error (MSE) as it takes bias and variance into account

$$MSE(T, H) = \frac{1}{n} ASV(T, H) + \text{Bias}(T, H) \text{Bias}(T, H)'. \quad (25)$$

Figure 12 displays MSE as a function of  $n$  for  $\beta_0 = 0.05$  and  $1.5$ ,  $\lambda = 0.1$  is fixed. We only present results for simple regression as they resemble the component-wise results in multiple regression.

Looking at Figure 12, the MSE of LS is the same in both panels as LS has no bias and its ASV does not depend on  $\beta_0$ . It decreases monotonously from one to zero. The MSEs of the other functionals are also monotonously decreasing, but towards their bias. For  $\beta_0 = 0.05$ , MSE of ridge is slightly lower than that of LS. The MSEs of the sparse functionals are constant and equal to their squared bias (i.e.  $\beta_0^2$  as the estimate equals zero). For  $\beta_0 = 1.5$ , MSE of biweight is largest, MSE of sparse LTS is slightly larger

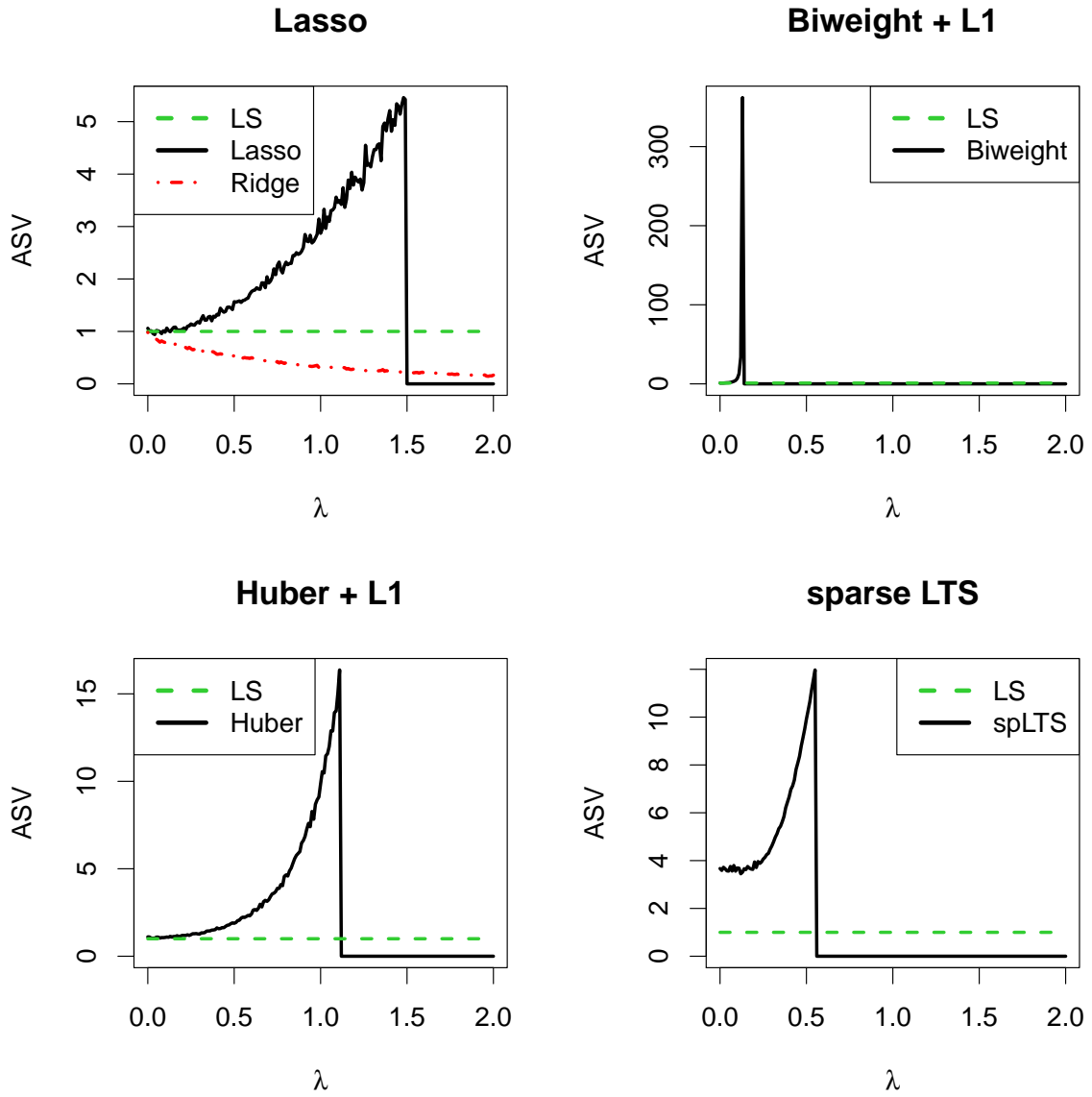


Figure 11: Asymptotic variance of various functionals for  $\beta_0 = 1.5$

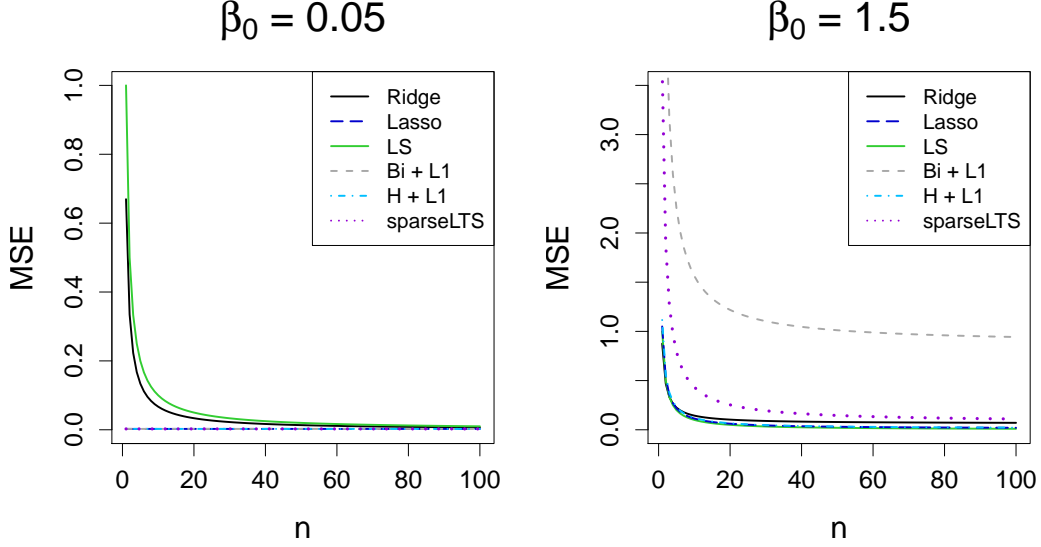


Figure 12: Mean squared error of various functionals ( $\lambda = 0.1$  fixed)

than ridge and MSE of the lasso and Huber is similar to LS, which is the lowest. We again do not show results for  $\beta_0 = 0$  because then no functional has a bias, and we would only compare the ASVs.

We also show the match at population and sample level for the MSE. For any estimator  $\hat{\beta}$  computed for  $r = 1, \dots, R$  samples, an estimator for the mean squared error (25) is

$$\widehat{MSE}(\hat{\beta}) = \frac{1}{R} \sum_{r=1}^R (\hat{\beta}_r - \beta_0)^2.$$

For the six functionals (LS, ridge, lasso, biweight-M with  $L_1$  penalty, Huber-M with  $L_1$  penalty and sparse LTS) used in this section, Figures 13 and 14 illustrate the good convergence of  $n \cdot \widehat{MSE}(\hat{\beta})$  to  $n \cdot MSE(\beta_0, F)$  for  $\beta_0 = 0.05$  and  $1.5$ , respectively.

## 10 Conclusion

In this paper we computed influence functions of penalized regression estimators, more precisely for penalized M-functionals. From the derivation of the influence function, we concluded that only functionals with a bounded loss function (biweight, sparse LTS) achieve robustness against leverage points, while a Huber loss can deal with vertical outliers. Looking at the MSE, sparse LTS is preferred in case of bad leverage points and the  $L_1$ -penalized Huber M-estimator in case there are only vertical outliers.

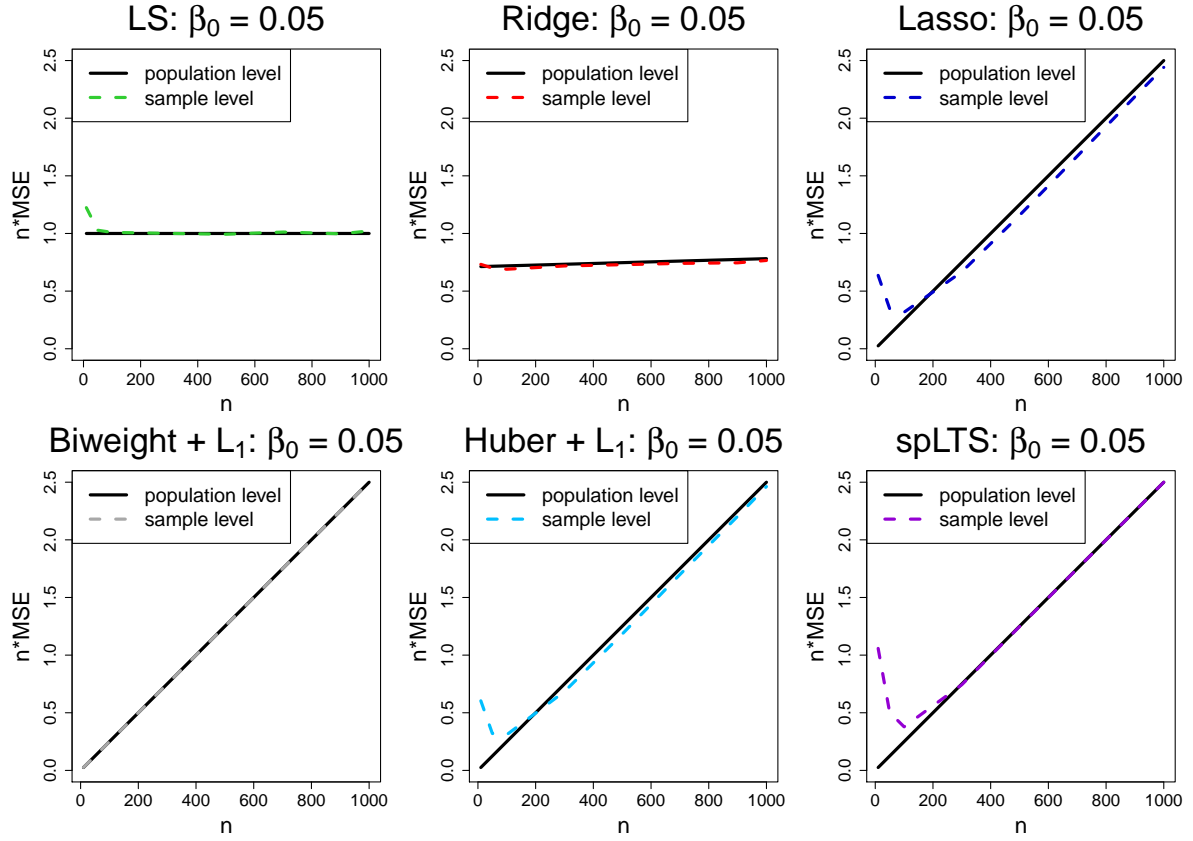


Figure 13: Convergence of  $\widehat{MSE}(\hat{\beta})$  to  $MSE(\beta_0, F)$  for different functionals with  $\beta_0 = 0.05$



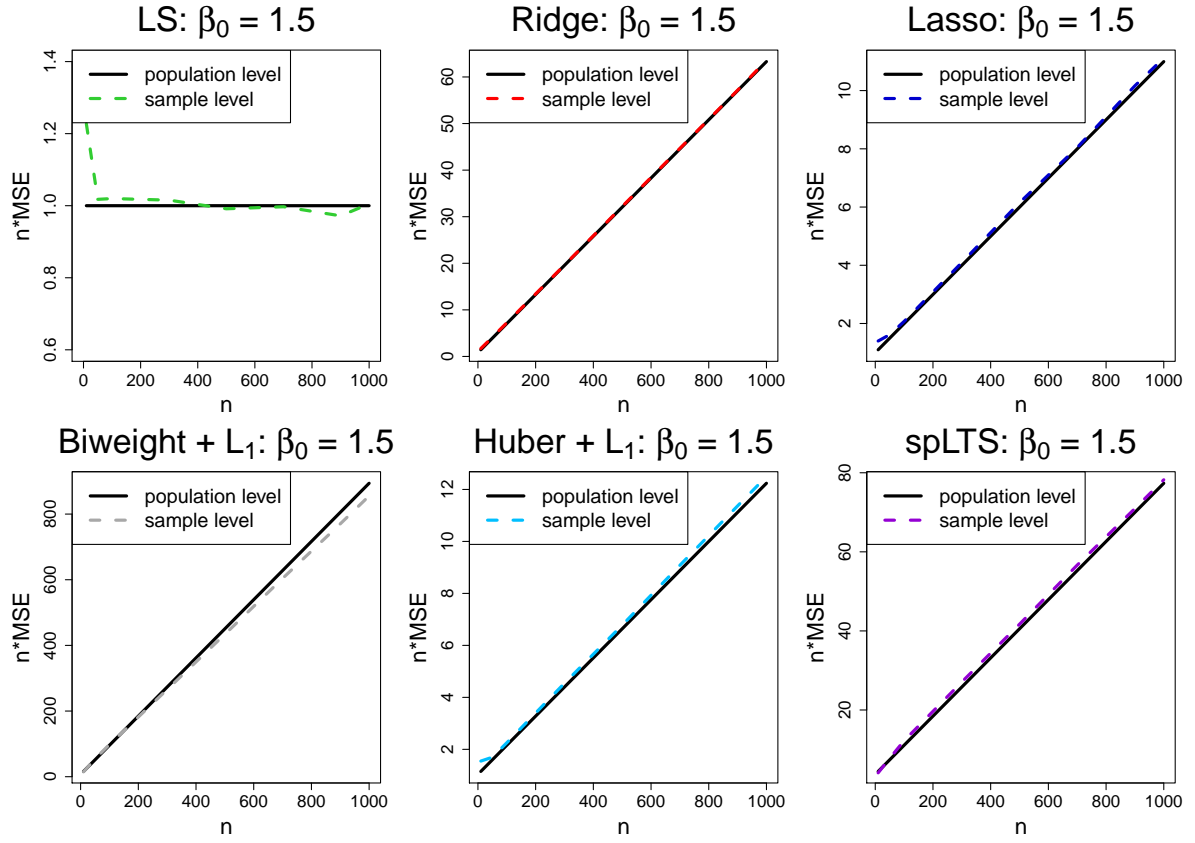


Figure 14: Convergence of  $\widehat{MSE}(\hat{\beta})$  to  $MSE(\beta_0, F)$  for different functionals with  $\beta_0 = 3$

Apart from considering the influence function, a suitable estimator is often also chosen with respect to its breakdown point [see for example Maronna et al., 2006]. This second important property in robust analysis gives the maximum fraction of outliers that a method can deal with. While it has already been computed for sparse LTS [see Alfons et al., 2013], it would also be worth deriving it for the other robust penalized M-functionals mentioned in this paper.

As any study, also this one is subject to some limitations. First of all, we assumed in our derivations the penalty parameter  $\lambda$  to be fixed. However, in practice it is often chosen with a data-driven approach. Thus, contamination in the data might also have an effect on the estimation through the choice of the penalty parameter. Investigation of this effect is left for further research.

Another limitation is that the values of the tuning constants in the loss functions of the M-estimators were selected to achieve a given efficiency in the non penalized case. One could imagine to select the  $\lambda$  parameter simultaneously with the other tuning constants.

Finally, in the theoretical derivations (but not at the sample level) we implicitly assume the scale of the error terms to be fixed, in order to keep the calculations feasible. While the results obtained for the lasso, the ridge and the sparse LTS functional do not rely on that assumption, the results for biweight and Huber loss do.

## APPENDIX - Proofs

*Proof of Equation 12.* Recall that we are in the case  $p = 1$ . For any joint distribution  $(x, y) \sim H$  with  $\beta_{LASSO}(H) \neq 0$ , minimizing the objective function in (7) and solving the resulting FOC for  $\beta_{LASSO}(H)$  yields

$$\beta_{LASSO}(H) = \beta_{LS}(H) - \frac{\lambda}{\mathbb{E}_H[x^2]} \text{sign}(\beta_{LASSO}(H)). \quad (26)$$

We will now consider two different cases. First we consider the case that the lasso functional is not zero at distribution  $H$ . We will show that it then always has to have the same sign as the LS functional  $\beta_{LS}(H)$ . We start with assuming  $\text{sign}(\beta_{LASSO}(H)) \neq \text{sign}(\beta_{LS}(H))$  and show that this will lead to a contradiction. In this case  $\beta_{LS}(H) = 0$  is not possible for the following reason. If  $\beta_{LS}(H) = 0$ , then  $\beta = 0$  minimizes the residual sum of squares. Furthermore, the minimum of the penalty function is attained at  $\beta = 0$ . Hence,  $\beta = 0$  would not only minimize the residual sum of squares, but also the penalized objective function, if  $\beta_{LS}(H) = 0$ . Hence, the lasso functional would also be zero, which we do not consider in this first case. Thus, take  $\beta_{LS}(H) > 0$ . From our assumption it would follow that  $\text{sign}(\beta_{LASSO}(H)) = -1$  (as  $\beta_{LASSO}(H) = 0$  is considered only in the next paragraph) and together with the FOC this would yield the contradiction  $0 > \beta_{LASSO}(H) = \beta_{LS}(H) + \lambda/\mathbb{E}_H[x^2] > \beta_{LS}(H) > 0$ . Analogous for  $\beta_{LS}(H) < 0$ . Hence, for  $\beta_{LASSO}(H) \neq 0$  the sign of the lasso and the LS functional are always equal.

Let's now consider the case where the lasso functional is zero at the distribution  $H$ . The FOC then makes use of the concept of subdifferentials [see Bertsekas, 1995] and can be written as  $|\beta_{LS}(H)| \leq \lambda/\mathbb{E}_H[x^2]$ . On the other hand, if  $|\beta_{LS}(H)| \leq \lambda/\mathbb{E}_H[x^2]$  assuming  $\beta_{LASSO}(H) \neq 0$  leads to a contradiction since equation (26) would imply that  $\text{sign}(\beta_{LASSO}(H)) = -\text{sign}(\beta_{LASSO}(H))$ . Thus, the lasso functional equals zero if and only if  $|\beta_{LS}(H)| \leq \lambda/\mathbb{E}_H[x^2]$ . Therefore the lasso functional for simple regression is (12).  $\square$

*Proof of Lemma 1.* As  $x$  and  $e$  are independent,  $y - x\beta$  is normally distributed  $y - x\beta \sim \mathcal{N}(0, \sigma^2 + (\beta_0 - \beta)^2 \Sigma)$  for any  $\beta \in \mathbb{R}$ . Defining  $\sigma^2(\beta) := \sigma^2 + (\beta_0 - \beta)^2 \Sigma$  we find  $q_\beta = \Phi^{-1}(\frac{\alpha+1}{2})\sigma(\beta)$ . We also introduce  $q_\alpha = \Phi^{-1}(\frac{\alpha+1}{2})$ . With this we can rewrite the

expected value of the objective function (11)

$$\begin{aligned}
\mathbb{E}_F[(y - x\beta)^2 I_{|y-x\beta| \leq q_\beta}] &= \sigma^2(\beta) \mathbb{E}_F\left[\frac{(y - x\beta)^2}{\sigma^2(\beta)} I_{\left[\frac{|y-x\beta|}{\sigma(\beta)} \leq q_\alpha\right]}\right] \\
&= \sigma^2(\beta) \mathbb{E}_Z[Z^2 I_{|Z| \leq q_\alpha}] \quad \text{with } Z \sim \mathcal{N}(0, 1) \\
&= \sigma^2(\beta)(-2q_\alpha\phi(q_\alpha) + \alpha).
\end{aligned} \tag{27}$$

Denoting  $c_1 := \alpha - 2q_\alpha\phi(q_\alpha)$ , we can say that

$$\beta_{spLTS}(F) = \arg \min_{\beta \in \mathbb{R}} c_1 \sigma^2(\beta) + \alpha \lambda |\beta|.$$

Separating into  $\beta \geq 0$  and  $\beta \leq 0$ , differentiating wrt  $\beta$  and setting the result to 0 gives Equation (14).  $\square$

*Proof of Proposition 1.* The objective function (6) is minimized by solving the first order condition (FOC), the derivative of the objective function set zero. At the contaminated model with distribution  $F_\epsilon := (1 - \epsilon)F + \epsilon \delta_{(\mathbf{x}_0, y_0)}$  this yields

$$-\mathbb{E}_{F_\epsilon}[\psi(y - \mathbf{x}'\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}] + 2\lambda J'(\boldsymbol{\beta}_M(F_\epsilon)) = 0.$$

Here  $J'(\boldsymbol{\beta}_M(F_\epsilon))$  is used as a shortcut for  $(J'(\beta_1(F_\epsilon)), \dots, J'(\beta_p(F_\epsilon)))'$  and  $\delta_{(\mathbf{x}_0, y_0)}$  denotes the point mass distribution at  $(\mathbf{x}_0, y_0)$ .

Using the definition of the contaminated distribution  $F_\epsilon$ , the FOC becomes

$$-(1 - \epsilon)\mathbb{E}_F[\psi(y - \mathbf{x}'\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}] - \epsilon\psi(y_0 - \mathbf{x}'_0\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}_0 + 2\lambda J'(\boldsymbol{\beta}_M(F_\epsilon)) = 0.$$

Derivation with respect to  $\epsilon$  yields

$$\begin{aligned}
&\mathbb{E}_F[\psi(y - \mathbf{x}'\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}] - (1 - \epsilon)\mathbb{E}_F[\psi'(y - \mathbf{x}'\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}(-\mathbf{x}'\frac{\partial}{\partial \epsilon}\boldsymbol{\beta}_M(F_\epsilon))] \\
&\quad - \psi(y_0 - \mathbf{x}'_0\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}_0 - \epsilon\psi'(y_0 - \mathbf{x}'_0\boldsymbol{\beta}_M(F_\epsilon))\mathbf{x}_0(-\mathbf{x}'_0\frac{\partial}{\partial \epsilon}\boldsymbol{\beta}_M(F_\epsilon)) \\
&\quad + 2\lambda \text{diag}(J''(\boldsymbol{\beta}_M(F_\epsilon)))\frac{\partial}{\partial \epsilon}\boldsymbol{\beta}_M(F_\epsilon) = 0,
\end{aligned}$$

where  $\text{diag}(J''(\boldsymbol{\beta}_M(F_\epsilon)))$  denotes the diagonal matrix with entries  $(J''((\beta_M(F_\epsilon))_1), \dots, J''((\beta_M(F_\epsilon))_p))$  in the main diagonal.

Since  $\frac{\partial}{\partial \epsilon}[\boldsymbol{\beta}_M(F_\epsilon)]|_{\epsilon=0} = IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_M, F)$ ,

$$\mathbb{E}_F[\psi(y - \mathbf{x}'\boldsymbol{\beta}_M(F))\mathbf{x}] + \mathbb{E}_F[\psi'(y - \mathbf{x}'\boldsymbol{\beta}_M(F))\mathbf{x}\mathbf{x}'] \cdot IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_M, F) \tag{28}$$

$$- \psi(y_0 - \mathbf{x}'_0\boldsymbol{\beta}_M(F))\mathbf{x}_0 + 2\lambda \text{diag}(J''(\boldsymbol{\beta}_M(F))) \cdot IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_M, F) = 0, \tag{29}$$

Solving (29) for  $IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_M, F)$ , gives Equation (15).  $\square$

*Proof of Lemma 2.* Using the explicit definition of the lasso functional (12), its influence function can be computed directly. Thus, we differentiate the functional at the contaminated model  $F_\epsilon = (1 - \epsilon)F + \epsilon\delta_{(x_0, y_0)}$  with respect to  $\epsilon$  and take the limit of  $\epsilon$  approaching 0

$$\begin{aligned}
IF((x_0, y_0), \beta_{LASSO}, F) &= \\
&= \frac{\partial}{\partial \epsilon} \left[ \text{sign}((1 - \epsilon)\mathbb{E}_F[xy] + \epsilon x_0 y_0) \left( \left| \frac{(1 - \epsilon)\mathbb{E}_F[xy] + \epsilon x_0 y_0}{(1 - \epsilon)\mathbb{E}_F[x^2] + \epsilon x_0^2} \right| - \frac{\lambda}{(1 - \epsilon)\mathbb{E}_F[x^2] + \epsilon x_0^2} \right)_+ \right] \Big|_{\epsilon=0} \\
&= \frac{\partial}{\partial \epsilon} [\text{sign}((1 - \epsilon)\mathbb{E}_F[xy] + \epsilon x_0 y_0)] \Big|_{\epsilon=0} \left( \left| \frac{\mathbb{E}_F[xy]}{\mathbb{E}_F[x^2]} \right| - \frac{\lambda}{\mathbb{E}_F[x^2]} \right)_+ + \\
&\quad + \text{sign}(\mathbb{E}_F[xy]) \frac{\partial}{\partial \epsilon} \left[ \left( \left| \frac{(1 - \epsilon)\mathbb{E}_F[xy] + \epsilon x_0 y_0}{(1 - \epsilon)\mathbb{E}_F[x^2] + \epsilon x_0^2} \right| - \frac{\lambda}{(1 - \epsilon)\mathbb{E}_F[x^2] + \epsilon x_0^2} \right)_+ \right] \Big|_{\epsilon=0}.
\end{aligned}$$

While the derivative in the first summand equals zero almost everywhere, the derivative occurring in the second summand has to consider two cases separately. Using the fact that  $\frac{\mathbb{E}_F[xy]}{\mathbb{E}_F[x^2]} = \beta_{LS}(F) = \beta_0$ , we get

$$\begin{aligned}
&\frac{\partial}{\partial \epsilon} \left[ \left( \left| \frac{(1 - \epsilon)\mathbb{E}_F[xy] + \epsilon x_0 y_0}{(1 - \epsilon)\mathbb{E}_F[x^2] + \epsilon x_0^2} \right| - \frac{\lambda}{(1 - \epsilon)\mathbb{E}_F[x^2] + \epsilon x_0^2} \right)_+ \right] \Big|_{\epsilon=0} = \\
&= \begin{cases} 0 & \text{if } -\frac{\lambda}{\mathbb{E}_F[x^2]} \leq \beta_0 < \frac{\lambda}{\mathbb{E}_F[x^2]} \\ \text{sign} \left( \frac{\mathbb{E}_F[xy]}{\mathbb{E}_F[x^2]} \right) \left( \frac{(-\mathbb{E}_F[xy] + x_0 y_0)\mathbb{E}_F[x^2] - \mathbb{E}_F[xy](-\mathbb{E}_F[x^2] + x_0^2)}{(\mathbb{E}_F[x^2])^2} \right) + \frac{\lambda(-\mathbb{E}_F[x^2] + x_0^2)}{(\mathbb{E}_F[x^2])^2} & \text{otherwise} \end{cases} \\
&= \begin{cases} 0 & \text{if } -\frac{\lambda}{\mathbb{E}_F[x^2]} \leq \beta_0 < \frac{\lambda}{\mathbb{E}_F[x^2]} \\ \text{sign}(\beta_0) \left( \frac{x_0(y_0 - \beta_0 x_0)}{\mathbb{E}_F[x^2]} \right) - \lambda \frac{\mathbb{E}_F[x^2] - x_0^2}{(\mathbb{E}_F[x^2])^2} & \text{otherwise.} \end{cases}
\end{aligned}$$

Thus, almost everywhere the influence function equals (17).  $\square$

*Proof of Lemma 3.* Differentiating the lasso functional of the coordinate descent algorithm

$$\beta_j^{cd}(H) = \text{sign} \left( \mathbb{E}_H \left[ x_j (y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)}) \right] \right) \left( \left| \frac{\mathbb{E}_H \left[ x_j (y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)}) \right]}{\mathbb{E}_H[x_j^2]} \right| - \frac{\lambda}{\mathbb{E}_H[x_j^2]} \right)_+$$

for the contaminated model  $(\mathbf{x}, y) \sim F_\epsilon = (1 - \epsilon)F + \epsilon\delta_{(\mathbf{x}_0, y_0)}$  yields

$$\begin{aligned}
IF((\mathbf{x}_0, y_0), \beta_j^{cd}, F, \boldsymbol{\beta}^*) &= \\
&= \frac{\partial}{\partial \epsilon} \left[ \text{sign} \left( \mathbb{E}_{F_\epsilon} \left[ x_j \left( y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)}(\epsilon) \right) \right] \right) \right] \Big|_{\epsilon=0} \left( \left| \frac{\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})]}{\mathbb{E}_F[x_j^2]} \right| - \frac{\lambda}{\mathbb{E}_F[x_j^2]} \right)_+ + \\
&\quad + \text{sign} \left( \mathbb{E}_F \left[ x_j \left( y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)} \right) \right] \right) \frac{\partial}{\partial \epsilon} \left[ \left( \left| \frac{\mathbb{E}_{F_\epsilon}[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)}(\epsilon))]}{\mathbb{E}_{F_\epsilon}[x_j^2]} \right| - \frac{\lambda}{\mathbb{E}_{F_\epsilon}[x_j^2]} \right)_+ \right] \Big|_{\epsilon=0}
\end{aligned} \tag{30}$$

Note that the fixed values  $\boldsymbol{\beta}^*(\epsilon)$  depend on  $\epsilon$ , as they may depend on the data, e.g. if they are the values of a previous coordinate descent loop.  $\boldsymbol{\beta}^{*(j)}$  is used as an abbreviation for  $\boldsymbol{\beta}^{*(j)}(0)$  and  $IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}^{*(j)}, F)$  is shortened to  $IF(\boldsymbol{\beta}^{*(j)})$ .

The derivative of the sign-function equals zero almost everywhere. For the derivation of the positive part function two different cases have to be considered

$$\begin{aligned}
\frac{\partial}{\partial \epsilon} \left[ \left( \left| \frac{(1 - \epsilon)\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)}(\epsilon))] + \epsilon(\mathbf{x}_0)_j(y_0 - \mathbf{x}_0^{(j)'} \boldsymbol{\beta}^{*(j)}(\epsilon))}{(1 - \epsilon)\mathbb{E}_F[x_j^2] + \epsilon(\mathbf{x}_0)_j^2} \right| - \frac{\lambda}{(1 - \epsilon)\mathbb{E}_F[x_j^2] + \epsilon(\mathbf{x}_0)_j^2} \right)_+ \right] \Big|_{\epsilon=0} &= \\
&= \begin{cases} 0 & \text{if } \left| \frac{\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})]}{\mathbb{E}_F[x_j^2]} \right| < \frac{\lambda}{\mathbb{E}_F[x_j^2]} \\ \text{sign} \left( \frac{\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})]}{\mathbb{E}_F[x_j^2]} \right) \left( \frac{(-\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})] + (-\mathbb{E}_F[x_j \mathbf{x}^{(j)'} IF(\boldsymbol{\beta}^{*(j)})]) + (\mathbf{x}_0)_j(y_0 - \mathbf{x}_0^{(j)'} \boldsymbol{\beta}^{*(j)}) \mathbb{E}_F[x_j^2]}{(\mathbb{E}_F[x_j^2])^2} \right. \\ \quad \left. + \frac{-\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})](-\mathbb{E}_F[x_j^2] + (\mathbf{x}_0)_j^2)}{(\mathbb{E}_F[x_j^2])^2} \right) - \frac{-\lambda(-\mathbb{E}_F[x_j^2] + (\mathbf{x}_0)_j^2)}{(\mathbb{E}_F[x_j^2])^2} & \text{otherwise} \end{cases} \\
&= \begin{cases} 0 & \\ \text{sign}(\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})]) \left( \frac{-\mathbb{E}_F[x_j \mathbf{x}^{(j)'} IF(\boldsymbol{\beta}^{*(j)})] + (\mathbf{x}_0)_j(y_0 - \mathbf{x}_0^{(j)'} \boldsymbol{\beta}^{*(j)})}{\mathbb{E}_F[x_j^2]} - \frac{\frac{\mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)})]}{\mathbb{E}_F[x_j^2]}(\mathbf{x}_0)_j^2}{\mathbb{E}_F[x_j^2]} \right) & \end{cases}
\end{aligned} \tag{31}$$

Using the result of equation (31) in (30) and denoting  $\tilde{y}^{(j)} := y - \mathbf{x}^{(j)'} \boldsymbol{\beta}^{*(j)}$  yields influence function (19).  $\square$

*Proof of Proposition 2.* W.l.o.g.  $\boldsymbol{\beta}_{LASSO} = (\tilde{\boldsymbol{\beta}}, 0, \dots, 0)'$  with  $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^k$  and  $\tilde{\beta}_j \neq 0 \forall j = 1, \dots, k$ . At first, we only consider variables  $j = 1, \dots, k$ . For them, the first order condition for finding the minimum of (7) yields

$$(-2\mathbb{E}_H[\mathbf{x}(y - \mathbf{x}' \boldsymbol{\beta}_{LASSO}(H))] + 2\lambda \text{sign}(\boldsymbol{\beta}_{LASSO}(H)))_j = 0 \quad j = 1, \dots, k$$

Let  $(\mathbf{x}, y) \sim F$  denote the model distribution and  $F_\epsilon$  the contaminated distribution. Then the FOC at the contaminated model is

$$-(1 - \epsilon)\mathbb{E}_F[x_j(y - \mathbf{x}'\boldsymbol{\beta}_{LASSO}(F_\epsilon))] - \epsilon(\mathbf{x}_0)_j(y - \mathbf{x}_0'\boldsymbol{\beta}_{LASSO}(F_\epsilon)) + \lambda \text{sign}((\boldsymbol{\beta}_{LASSO}(F_\epsilon))_j) = 0$$

After differentiating with respect to  $\epsilon$ , we get

$$\begin{aligned} \mathbb{E}_F[x_j(y - \mathbf{x}'\boldsymbol{\beta}_{LASSO}(F_\epsilon))] + (1 - \epsilon) \left( \mathbb{E}_F[x_j\mathbf{x}'] \frac{\partial \boldsymbol{\beta}_{LASSO}(F_\epsilon)}{\partial \epsilon} \right) - \\ - (\mathbf{x}_0)_j(y - \mathbf{x}_0'\boldsymbol{\beta}_{LASSO}(F_\epsilon)) + \epsilon(\mathbf{x}_0)_j \left( \mathbf{x}_0' \frac{\partial \boldsymbol{\beta}_{LASSO}(F_\epsilon)}{\partial \epsilon} \right) = 0. \end{aligned}$$

Taking the limit as  $\epsilon$  approaches 0 gives an implicit definition of the IF for  $j = 1, \dots, k$

$$\mathbb{E}_F[x_j\mathbf{x}'] \cdot IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_{LASSO}, F) = (\mathbf{x}_0)_j(y - \mathbf{x}_0'\boldsymbol{\beta}_{LASSO}(F)) - \mathbb{E}_F[x_j(y - \mathbf{x}'\boldsymbol{\beta}_{LASSO}(F))] \quad (32)$$

For variables  $j = k + 1, \dots, p$  with  $(\boldsymbol{\beta}_{LASSO})_j = 0$ , we need to use subgradients [see Bertsekas, 1995] to get the first order condition

$$\mathbf{0} \in -\mathbb{E}_H[\mathbf{x}(y - \mathbf{x}'\boldsymbol{\beta}_{LASSO}(H))] + \lambda \cdot \partial(\|\boldsymbol{\beta}_{LASSO}(H)\|_1)$$

Observing each variable individually yields

$$|\mathbb{E}_H[x_j(y - \mathbf{x}'\boldsymbol{\beta}_{LASSO}(H))]| \leq \lambda. \quad (33)$$

The coordinate descent algorithm converges for any starting value  $\boldsymbol{\beta}^*$  to  $\boldsymbol{\beta}_{LASSO}$  [see Friedman et al., 2007; Tseng, 2001], i.e. after enough updates  $\boldsymbol{\beta}^* \approx \boldsymbol{\beta}_{LASSO}$ . Thus, for  $(\boldsymbol{\beta}_{LASSO}(F))_j = 0$  and  $(\mathbf{x}, y) \sim F$ , Equation (33) yields

$$\left| \mathbb{E}_F[x_j(y - \mathbf{x}^{(j)'}\boldsymbol{\beta}^{*(j)})] \right| \leq \lambda.$$

Lemma 3 tells us then that

$$IF((\mathbf{x}_0, y_0), (\boldsymbol{\beta}_{LASSO})_j, F) = 0 \quad \forall j = k + 1, \dots, p.$$

With this we can rewrite Equation (32) as

$$\mathbb{E}_F[\mathbf{x}_{1:k}\mathbf{x}_{1:k}'] \cdot IF((\mathbf{x}_0, y_0), (\boldsymbol{\beta}_{LASSO})_{1:k}, F) = (\mathbf{x}_0)_{1:k}(y - \mathbf{x}_0'\boldsymbol{\beta}_{LASSO}(F)) - \mathbb{E}_F[x_{1:k}(y - \mathbf{x}'\boldsymbol{\beta}_{LASSO}(F))]$$

Multiplying with  $\mathbb{E}_F[\mathbf{x}_{1:k}\mathbf{x}_{1:k}']^{-1}$  from the left side, we get the influence function of the lasso functional (20).  $\square$

*Proof of Lemma 4.* We apply Proposition 1 with a quadratic loss function and use the second derivative of the penalty function  $J_K$

$$J_K''((\boldsymbol{\beta}_K)_j) = \begin{cases} J_K''((\boldsymbol{\beta}_K)_j) =: a_j & j = 1, \dots, k \\ 2K & j = k+1, \dots, p. \end{cases}$$

Wlog we take  $\sigma = 1$ . This gives the influence function of  $\boldsymbol{\beta}_K(F)$

$$IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_K, F) = (\mathbb{E}_F[\mathbf{x}\mathbf{x}'] + \lambda \text{diag}(J_K''((\boldsymbol{\beta}_K)_1), \dots, J_K''((\boldsymbol{\beta}_K)_k), 2K, \dots, 2K))^{-1} \cdot ((y_0 - \mathbf{x}_0' \boldsymbol{\beta}_K(F)) \mathbf{x}_0 - \mathbb{E}_F[(y - \mathbf{x}' \boldsymbol{\beta}_K(F)) \mathbf{x}])$$

The covariance matrix  $\mathbb{E}_F[\mathbf{x}\mathbf{x}']$  can be denoted as a block matrix

$$\mathbb{E}_F[\mathbf{x}\mathbf{x}'] = \begin{pmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{pmatrix}.$$

The inverse matrix needed in the IF is then

$$\begin{aligned} & (\mathbb{E}_F[\mathbf{x}\mathbf{x}'] + \lambda \text{diag}(J_K''((\boldsymbol{\beta}_K)_1), \dots, J_K''((\boldsymbol{\beta}_K)_k), 2K, \dots, 2K))^{-1} = \\ & = \begin{pmatrix} E_{11} + \lambda \text{diag}(J_K''((\boldsymbol{\beta}_K)_{1:k})) & E_{12} \\ E_{21} & E_{22} + 2\lambda K I_{p-k} \end{pmatrix}^{-1}. \end{aligned} \quad (34)$$

The inverse of the block matrix can be computed as

$$(\mathbb{E}_F[\mathbf{x}\mathbf{x}'] + \lambda \text{diag}(0, \dots, 0, 2K, \dots, 2K))^{-1} = \begin{pmatrix} A^{-1} + A E_{12} C^{-1} E_{21} A^{-1} & -A^{-1} E_{12} C^{-1} \\ -C^{-1} E_{21} A^{-1} & C^{-1} \end{pmatrix}$$

with  $C = E_{22} + 2\lambda K I_{p-k} - E_{21} A^{-1} E_{12}$  and  $A = E_{11} + \lambda \text{diag}(J_K''((\boldsymbol{\beta}_K)_{1:k}))$  [see Magnus and Neudecker, 2002, p11].

We denote the eigenvalues of matrix  $D = E_{22} - E_{21} E_{11}^{-1} E_{12}$  by  $\nu_1, \dots, \nu_{p-k}$ . Then the eigenvalues of the symmetric positive definite matrix  $C$  are  $\nu_1 + 2\lambda K, \dots, \nu_{p-k} + 2\lambda K$ . If  $K$  approaches infinity, these eigenvalues also tend to infinity. Hence, all eigenvalues of  $C^{-1}$  converge to zero. Thus,  $C^{-1}$  becomes the zero matrix and therefore the inverse matrix in (34) converges to

$$\lim_{K \rightarrow \infty} (\mathbb{E}_F[\mathbf{x}\mathbf{x}'] + \lambda \text{diag}(0, \dots, 0, 2K, \dots, 2K))^{-1} = \begin{pmatrix} E_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

This gives the IF of the lasso functional (20) as the limit of  $IF((\mathbf{x}_0, y_0), \boldsymbol{\beta}_K, F)$  for  $K \rightarrow \infty$ .  $\square$



*Proof of Lemma 5.* As the sparse LTS functional is continuous, the influence function of the sparse LTS functional equals 0 if  $\beta_{spLTS}(F) = 0$ . Thus, assume from now on  $\beta_{spLTS}(F) \neq 0$ .

The first order condition at the contaminated model  $F_\epsilon = (1 - \epsilon)F + \epsilon\delta_{(x_0, y_0)}$  yields

$$0 = \frac{\partial}{\partial \beta} \left( \int_{-q_{\epsilon, \beta}}^{q_{\epsilon, \beta}} u^2 dF_\epsilon^\beta(u) \right) + \alpha \lambda \text{sign}(\beta) =: \Psi(\epsilon, \beta). \quad (35)$$

Note that here the quantile  $q_{\epsilon, \beta}$  as well as the joint model distribution of  $x$  and  $y$   $F_\epsilon^\beta$  depend on  $\beta$ . We denote the solution of (35) by  $\beta_\epsilon := \beta_{spLTS}(F_\epsilon)$  for  $\beta \neq 0$  and  $\beta_{spLTS}(F)$  otherwise.

As (35) is true for all  $\epsilon \in \mathbb{R}_+$ , the chain rule gives

$$\begin{aligned} 0 &= \frac{\partial}{\partial \epsilon} [\Psi(\epsilon, \beta_\epsilon)]|_{\epsilon=0} = \Psi_1(0, \beta_{spLTS}(F)) + \Psi_2(0, \beta_{spLTS}(F)) IF(\beta_{spLTS}) \\ &\rightsquigarrow IF(\beta_{spLTS}) = -[\Psi_2(0, \beta_{spLTS}(F))]^{-1} \Psi_1(0, \beta_{spLTS}(F)) \end{aligned} \quad (36)$$

where  $\Psi_1(a, b) = \frac{\partial}{\partial \epsilon} \Psi(\epsilon, b)|_{\epsilon=a}$  and  $\Psi_2(a, b) = \frac{\partial}{\partial \beta} \Psi(a, \beta)|_{\beta=b}$ .

Before computing  $\Psi_1(0, \beta_{spLTS}(F))$  and  $\Psi_2(0, \beta_{spLTS}(F))$ , we can simplify  $\Psi(\epsilon, \beta)$  by using  $F^\beta = \mathcal{N}(0, \sigma^2(\beta))$  with  $\sigma^2(\beta) = \sigma^2 + (\beta_{spLTS}(F) - \beta)^2 \Sigma$

$$\begin{aligned} \Psi(\epsilon, \beta) &= \frac{\partial}{\partial \beta} \left( (1 - \epsilon) \int_{-q_{\epsilon, \beta}}^{q_{\epsilon, \beta}} u^2 dF^\beta(u) + \epsilon I_{[|y_0 - x_0 \beta| \leq q_{\epsilon, \beta}]} (y_0 - x_0 \beta)^2 \right) + \alpha \lambda \text{sign}(\beta) \\ &= (1 - \epsilon) \frac{\partial}{\partial \beta} \left( \int_{-q_{\epsilon, \beta}}^{q_{\epsilon, \beta}} \frac{u^2}{\sigma(\beta)} \phi\left(\frac{u}{\sigma(\beta)}\right) du \right) - 2\epsilon x_0 (y_0 - x_0 \beta) I_{[|y_0 - x_0 \beta| \leq q_{\epsilon, \beta}]} + \alpha \lambda \text{sign}(\beta) \end{aligned}$$

and the Leibniz integral rule

$$\begin{aligned} \frac{\partial}{\partial \beta} \left( \int_{-q_{\epsilon, \beta}}^{q_{\epsilon, \beta}} \frac{u^2}{\sigma(\beta)} \phi\left(\frac{u}{\sigma(\beta)}\right) du \right) &= \int_{-q_{\epsilon, \beta}}^{q_{\epsilon, \beta}} u^2 \phi\left(\frac{u}{\sigma(\beta)}\right) \left(1 - \frac{u^2}{\sigma^2(\beta)}\right) du \frac{(\beta_0 - \beta)\Sigma}{\sigma^3(\beta)} + \\ &\quad + 2 \frac{q_{\epsilon, \beta}^2}{\sigma(\beta)} \phi\left(\frac{q_{\epsilon, \beta}}{\sigma(\beta)}\right) \frac{\partial}{\partial \beta} [q_{\epsilon, \beta}]. \end{aligned}$$

To obtain the derivative  $\Psi_1(0, \beta_{spLTS}(F))$ , we can again use the Leibniz integral rule

$$\begin{aligned}
\Psi_1(0, \beta_{spLTS}(F)) = & - \left( \int_{-q_0, \beta_{spLTS}(F)}^{q_0, \beta_{spLTS}(F)} u^2 \phi\left(\frac{u}{\sigma(\beta_{spLTS}(F))}\right) \left(1 - \frac{u^2}{\sigma^2(\beta_{spLTS}(F))}\right) du \frac{(\beta_0 - \beta_{spLTS}(F))\Sigma}{\sigma^3(\beta_{spLTS}(F))} + \right. \\
& \left. + 2 \frac{q_{0, \beta_{spLTS}(F)}^2}{\sigma(\beta_{spLTS}(F))} \phi\left(\frac{q_{0, \beta_{spLTS}(F)}}{\sigma(\beta_{spLTS}(F))}\right) \frac{\partial}{\partial \beta} [q_{0, \beta}]|_{\beta=\beta_{spLTS}(F)} \right) + \\
& + \frac{\partial}{\partial \epsilon} \left[ \int_{-q_{\epsilon, \beta_{spLTS}(F)}}^{q_{\epsilon, \beta_{spLTS}(F)}} u^2 \phi\left(\frac{u}{\sigma(\beta_{spLTS}(F))}\right) \left(1 - \frac{u^2}{\sigma^2(\beta_{spLTS}(F))}\right) du \right] \Big|_{\epsilon=0} \frac{(\beta_0 - \beta_{spLTS}(F))\Sigma}{\sigma^3(\beta_{spLTS}(F))} + \\
& + 4 \frac{q_{0, \beta_{spLTS}(F)}}{\sigma(\beta_{spLTS}(F))} \frac{\partial}{\partial \epsilon} [q_{\epsilon, \beta_{spLTS}(F)}]|_{\epsilon=0} \phi\left(\frac{q_{0, \beta_{spLTS}(F)}}{\sigma(\beta_{spLTS}(F))}\right) \frac{\partial}{\partial \beta} [q_{0, \beta}]|_{\beta=\beta_{spLTS}(F)} + \\
& + 2 \frac{q_{0, \beta_{spLTS}(F)}^2}{\sigma(\beta_{spLTS}(F))} \phi'\left(\frac{q_{0, \beta_{spLTS}(F)}}{\sigma(\beta_{spLTS}(F))}\right) \frac{\partial}{\partial \epsilon} [q_{\epsilon, \beta_{spLTS}(F)}]|_{\epsilon=0} \frac{1}{\sigma(\beta_{spLTS}(F))} \frac{\partial}{\partial \beta} [q_{0, \beta}]|_{\beta=\beta_{spLTS}(F)} + \\
& + 2 \frac{q_{0, \beta_{spLTS}(F)}^2}{\sigma(\beta_{spLTS}(F))} \phi\left(\frac{q_{0, \beta_{spLTS}(F)}}{\sigma(\beta_{spLTS}(F))}\right) \frac{\partial}{\partial \epsilon} \left[ \frac{\partial}{\partial \beta} [q_{\epsilon, \beta}]|_{\beta=\beta_{spLTS}(F)} \right] \Big|_{\epsilon=0} - \\
& - 2x_0(y_0 - x_0\beta_{spLTS}(F)) I_{[|y_0 - x_0\beta_{spLTS}(F)| \leq q_{0, \beta_{spLTS}(F)}]}.
\end{aligned}$$

The derivatives of the quantiles can be derived from equations  $\bar{F}_{\epsilon}^{\beta}(q_{\epsilon}, \beta) = \alpha$  and  $\bar{F}^{\beta}(q_0, \beta) = \alpha$  by differentiating wrt the required variables. This yields

$$\begin{aligned}
\frac{\partial}{\partial \epsilon} [q_{\epsilon, \beta_{spLTS}(F)}]|_{\epsilon=0} &= \frac{\alpha - I_{[|y_0 - x_0\beta_{spLTS}(F)| \leq q_{0, \beta_{spLTS}(F)}]}}{2\phi(q_{\alpha}) \frac{1}{\sigma(\beta_{spLTS}(F))}} \\
\frac{\partial}{\partial \beta} [q_{0, \beta}]|_{\beta=\beta_{spLTS}(F)} &= - \frac{q_{0, \beta_{spLTS}(F)}(\beta_0 - \beta_{spLTS}(F))\Sigma}{\sigma^2(\beta_{spLTS}(F))} \\
\frac{\partial}{\partial \epsilon} \left[ \frac{\partial}{\partial \beta} [q_{\epsilon, \beta}]|_{\beta=\beta_{spLTS}(F)} \right] \Big|_{\epsilon=0} &= \frac{I_{[|r_0| \leq q_{\alpha}]} - \alpha}{2\phi(q_{\alpha})} \cdot \frac{(\beta_0 - \beta_{spLTS}(F))\Sigma}{\sigma(\beta_{spLTS}(F))}
\end{aligned}$$

with  $r_0 := \frac{y_0 - x_0\beta_{spLTS}(F)}{\sigma(\beta_{spLTS}(F))}$ .

Thus,

$$\Psi_1(0, \beta_{spLTS}(F)) = (-4q_{\alpha}\phi(q_{\alpha}) + 2\alpha + 2q_{\alpha}^2(I_{[|r_0| \leq q_{\alpha}]} - \alpha))(\beta_0 - \beta_{spLTS}(F))\Sigma \quad (37)$$

$$- 2x_0(y_0 - x_0\beta_{spLTS}(F)) I_{[|r_0| \leq q_{\alpha}]}. \quad (38)$$

With similar ideas as in the derivation of  $\Psi_1(0, \beta_{spLTS}(F))$ , we get

$$\Psi_2(0, \beta_{spLTS}(F)) = (-4q_{\alpha}\phi(q_{\alpha}) + 4\Phi(q_{\alpha}) - 2)\Sigma. \quad (39)$$

Using (38) and (39) in (36), we get the influence function (23) of the sparse LTS functional for simple regression.  $\square$

## References

- A. Alfons, C. Croux, and S. Gelper. Sparse least trimmed squares regression for analyzing high-dimensional large data sets. *The Annals of Applied Statistics*, 7(1):226–248, 2013.
- O. Arslan. Weighted LAD-lasso method for robust parameter estimation and variable selection in regression. *Computation Statistics and Data Analysis*, 56(6):1952–1965, 2012.
- B. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.
- J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.
- J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.
- W. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7(3):397–416, 1998.
- F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics. The Approach based on Influence Functions*. John Wiley & Sons, New York, 1986.
- A.E. Hoerl and R.W. Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1977.
- G. Li, H. Peng, and L. Zhu. Nonconcave penalized M-estimation with a diverging number of parameters. *Statistica Sinica*, 21:391–419, 2011.
- J.R. Magnus and H. Neudecker. *Matrix differential calculus with applications in Statistics and Econometrics*. John Wiley & Sons, 2nd edition, 2002.
- R.A. Maronna, R.D. Martin, and V.J. Yohai. *Robust Statistics*. John Wiley & Sons, Hoboken, New Jersey, 2nd edition, 2006.
- M.R. Osborne. *Finite Algorithms in Optimization and Data Analysis*. John Wiley & Sons, Chichester, 1985.
- P.J. Rousseeuw and A.M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.
- P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.
- H. Wang, G. Li, and G. Jiang. Robust regression shrinkage and consistent variable selection through the LAD-lasso. *Journal of Business & Economic Statistics*, 25(3):347–355, 2007.

**FACULTY OF ECONOMICS AND BUSINESS**

Naamsestraat 69 bus 3500

3000 LEUVEN, BELGIË

tel. + 32 16 32 66 12

fax + 32 16 32 67 91

info@econ.kuleuven.be

www.econ.kuleuven.be

